1998-09

# Design of a microelectronic controller with a MIL-STD-1553 bus interface for the tactile situation awareness system

Luke, Brian L.

Monterey, California. Naval Postgraduate School

# NAVAL POSTGRADUATE SCHOOL
# MONTEREY, CALIFORNIA


# THESIS

**DESIGN OF A MICROELECTRONIC CONTROLLER WITH A MIL-STD-1553 BUS INTERFACE FOR THE TACTILE SITUATION AWARENESS SYSTEM**

by

Brian L. Luke

September 1998

Thesis Advisor: Douglas J. Fouts

**Approved for public release; distribution is unlimited.**

# REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188



| 1. AGENCY USE ONLY *(Leave blank)* | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | September 1998 | Engineer's Thesis |

| | |
|---|---|
| 4. TITLE AND SUBTITLE DESIGN OF A MICROELECTRONIC CONTROLLER WITH A MIL-STD-1553 BUS INTERFACE FOR THE TACTILE SITUATION AWARENESS SYSTEM | 5. FUNDING NUMBERS |
| 6. AUTHOR(S) Brian L. Luke | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)<br>Naval Postgraduate School<br>Monterey, CA 93943-5000 | 8. PERFORMING ORGANIZATION REPORT NUMBER |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | 10. SPONSORING/MONITORING AGENCY REPORT NUMBER |

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

| | |
|---|---|
| 12a. DISTRIBUTION/AVAILABILITY STATEMENT<br>Approved for public release; distribution is unlimited. | 12b. DISTRIBUTION CODE |

13. ABSTRACT *(maximum 200 words)*

Spatial Disorientation (SD) is a triservice aviation problem that costs the Department of Defense more than $300 million annually in destroyed aircraft and is the primary cause of pilot-related mishaps in the Navy and the Air Force. As one solution to the SD problem, the Naval Aerospace Medical Research Laboratory has developed the Tactile Situation Awareness System (TSAS). The primary objective of TSAS is to enhance pilot performance and reduce SD-related aircrew/aircraft losses by providing continuous non-visual information using the normally underutilized sensory channel of touch. Using vibrotactile stimulators, TSAS applies information taken from the aircraft's instruments to the pilot's torso. The current implementation of TSAS is a research system that is not compatible with the crowded cockpit of modern aircraft. This thesis presents a design of a microelectronic controller for TSAS compatible with tactical environments. This new system, called the Tactor Interface Microcontroller System (TIMS), incorporates the functionality of the research TSAS into a palm-sized microcontroller system and enables TSAS to communicate directly to the computerized sensory and weapons systems in combat aircraft such as the Navy F/A-18. TIMS brings the TSAS prototype out of the research stage and puts this exciting technology into the hands of the warfighter.

| | | | |
|---|---|---|---|
| 14. SUBJECT TERMS Electronics, Humai :ms Interface, TSAS, Embedded System | | | 15. NUMBER OF PAGES 180 |
| | | | 16. PRICE CODE |
| 17. SECURITY CLASSIFICATION OF REPORT<br>Unclassified | 18. SECURITY CLASSIFICATION OF THIS PAGE<br>Unclassified | 19. SECURITY CLASSIFICATION OF ABSTRACT<br>Unclassified | 20. LIMITATION OF ABSTRACT<br>UL |

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

**Approved for public release; distribution is unlimited.**

# DESIGN OF A MICROELECTRONIC CONTROLLER WITH A MIL-STD-1553 BUS INTERFACE FOR THE TACTILE SITUATION AWARENESS SYSTEM

Brian L. Luke
Lieutenant, United States Navy
B.S., United States Naval Academy, 1992

Submitted in partial fulfillment
of the requirements for the degree of

**ELECTRICAL ENGINEER**

from the

**NAVAL POSTGRADUATE SCHOOL**
**September 1998**

## ABSTRACT

Spatial Disorientation (SD) is a triservice aviation problem that costs the Department of Defense more than $300 million annually in destroyed aircraft and is the primary cause of pilot-related mishaps in the Navy and the Air Force. As one solution to the SD problem, the Naval Aerospace Medical Research Laboratory has developed the Tactile Situation Awareness System (TSAS). The primary objective of TSAS is to enhance pilot performance and reduce SD-related aircrew/aircraft losses by providing continuous non-visual information using the normally underutilized sensory channel of touch. Using vibrotactile stimulators, TSAS applies information taken from the aircraft's instruments to the pilot's torso. The current implementation of TSAS is a research system that is not compatible with the crowded cockpit of modern aircraft. This thesis presents a design of a microelectronic controller for TSAS compatible with tactical environments. This new system, called the Tactor Interface Microcontroller System (TIMS), incorporates the functionality of the research TSAS into a palm-sized microcontroller system and enables TSAS to communicate directly to the computerized sensory and weapons systems in combat aircraft such as the Navy F/A-18. TIMS brings the TSAS prototype out of the research stage and puts this exciting technology into the hands of the warfighter.

## TABLE OF CONTENTS

## LIST OF FIGURES

## LIST OF SYMBOLS, ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| 1553 | see MIL-STD-1553 |
| AC | Auxiliary Command |
| ACE | Advanced Communication Engine |
| ADS | Application Development System |
| ASIC | Application Specific Integrated Circuit |
| ATM | Asynchronous Transfer Mode |
| BC | Bus Controller |
| BER | Bit Error Rate |
| BGA | Ball Grid Array |
| BISYNC | Binary Synchronous Communications |
| bps | bits per second |
| BRG | Baud Rate Generator |
| CD | Carrier Detect |
| CISC | Complex Instruction Set Computer |
| COTS | Commercial off the Shelf |
| CP | Communications Processor |
| CPIC | CPM Interrupt Controller |
| CPLD | Complex Programmable Logic Device |
| CPM | Communication Processor Module |
| CPU | Central Processing Unit |
| CRC | Cyclic Redundancy Check |
| CTS | Clear To Send |
| DC | Direct Current |
| Dff | D-Flip-Flop |
| DIP | Dual In-line Package |
| DMA | Direct Memory Access |
| DoD | Department of Defense |

| | |
|---|---|
| DPLL | Digital Phase Lock Loop |
| DRAM | Dynamic Random Access Memory |
| DSP | Digital Signal Processing |
| EDO RAM | Extended Data Out RAM |
| EEPROM | Electronically EPROM |
| EI | Excitation Interval |
| EPROM | Erasable Programmable Read Only Memory |
| EVA | Extra Vehicular Activity |
| F | Farads |
| FPGA | Field Programmable Gate Array |
| GCA | Ground Controlled Approach |
| GPS | Global Positioning System |
| H | Henries |
| HDLC | High-Level Data Link Control |
| Hz | Hertz |
| I/O | Input and Output |
| $I^2C$ | Inter-integrated Controller |
| IC | Integrated Circuit |
| IDMA | Independent DMA |
| IEEE | Institute of Electrical and Electronics Engineers |
| IMC | Instrument Meteorological Conditions |
| IrDA | Infrared Data Association |
| ISDN | Integrated Services Digital Network |
| JSF | Joint Strike Fighter |
| LAN | Local Area Network |
| LED | Light Emitting Diode |
| MAC | Multiply and Accumulate |
| MCM | Multi-Chip Module |
| MIL-STD-1553 | Military Standard Specification 1553 Aircraft Internal Data Bus |

| | |
|---|---|
| MIPS | Million Instructions Per Second |
| MMU | Memory Management Unit |
| MPC860 | Motorola MPC860 PowerQUICC |
| MT | Monitor Terminal |
| NAMRL | Naval Aerospace Medical Research Laboratory |
| NPS | Naval Postgraduate School |
| NRZ | Not Return to Zero |
| NRZI | Not Return to Zero Inverted |
| OSI | Open Systems Interconnection |
| OTP | One Time Programmable |
| PBGA | Plastic BGA |
| PC | Personal Computer |
| PCB | Printed Circuit Board |
| PCMCIA | Personal Computer Memory Card International Association |
| PGA | Pin Grid Array |
| PIT | Periodic Interrupt Timer |
| PLL | Phase Lock Loop |
| PowerQUICC | PowerPC Quad Integrated Communications Controller |
| QFP | Quad Flat Pack |
| QUICC | Quad Integrated Communications Controller |
| RAM | Random Access Memory |
| RISC | Reduced Instruction Set Computer |
| ROM | Read Only Memory |
| RP | Repetition Period |
| RPV | Remotely Piloted Vehicle |
| RT | Remote Terminal |
| RTS | Request To Send |
| SA | Situation Awareness |
| SCC | Serial Communications Controller |

| | |
|---|---|
| SD | Spatial Disorientation |
| SDMA | Serial DMA |
| SIMM | Single In-line Memory Module |
| SINS | Seal Inshore Navigation System |
| SIU | System Interface Unit |
| SMC | Serial Management Controller |
| SPI | Serial Peripheral Interface |
| SPLD | Simple Programmable Logic Device |
| SRAM | Static RAM |
| TDM | Time-Division Multiplexing |
| TIC | Tactor Interface Chip |
| TIMS | Tactor Interface Microcontroller System |
| TLS | Tactor Locator System |
| TSAS | Tactile Situation Awareness System |
| TSOP | Thin Small Outline Package |
| TTL | Transistor-Transistor Logic |
| UART | Universal Asynchronous Receiver Transmitter |
| USART | Synchronous Universal Asynchronous Receiver Transmitter |
| UUV | Unmanned Underwater Vehicle |
| UV | Ultraviolet |
| V | Volts |
| VLSI | Very Large Scale Integration |
| VMC | Visual Meteorological Conditions |
| VSWMCM | Very Shallow Water Mine Countermeasures |
| WAN | Wide Area Network |

## ACKNOWLEDGEMENT

I would like to acknowledge the financial support of the Naval Aerospace Medical Research Laboratory (NAMRL), Spatial Orientation Systems Department, for allowing the purchase of equipment used in this thesis. Sincere appreciation is extended to Commander Angus Rupert and Braden McGrath of NAMRL for technical, professional, and personal guidance throughout this entire project.

I also would like to thank Professor Douglas Fouts for his willingness to sponsor a new thesis topic to the Electrical and Computer Engineering Department and for his continuously open-minded and creative guidance with this thesis. Furthermore, I want to express a genuine gratitude to Ron Phelps of the Naval Postgraduate School for his weekly tutorials on printed circuit board design. Lastly, this thesis would not have been possible without the help, support, and editing ability of my wife, Heather.

This thesis and all my work on this project is dedicated to my son, Austin. His love of airplanes and flying fuels my hope that he will have a system like the one detailed in this document to keep him safe when he eventually takes to the air.

# I. INTRODUCTION

In the words of Leonardo da Vinci, "Human subtlety . . . will never devise an invention more beautiful, more simple, or more direct than does nature, because in her inventions nothing is lacking and nothing is superfluous." Human-factors researchers in the field of Naval Aviation are well aware of this quotation, whereas aeronautical engineers possibly are not. In man's quest to fly, aircraft designers have taken several lessons from those animals in nature with centuries of experience in practical aviation: the birds. Examining the picture of a Gannet in slow flight, Figure 1, top, one can see many similarities to modern aircraft in the U.S. military inventory, like the Navy F-14, Figure 1, bottom, approaching an aircraft carrier for landing. The Gannet's body is cambered and its tail is down to reduce speed and increase lift. Trim is accomplished by sweeping his wings forward to displace the center of pressure ahead of the center of gravity. As speed is reduced, airflow begins to separate near the wing trailing edges. This causes soft feathers to ruffle upwards, turbulating the boundary layer, delaying further separation and wing stall. The Alula, or bastard wing, acts as a slat by extending at the leading edge to delay stall by keeping airflow attached. Lastly, the landing gear is lowered to increase drag. [Ref. 1]

The similarities between today's modern aircraft and our feathered friends are numerous, but engineers still have much to learn from nature in the field of aviation. Aeronautical engineers may have copied the body of a bird, but not adequate means to control it. In the cockpits of today's modern aircraft is a dizzying array of computer

displays, lights, and sounds which attempt to convey information to the pilot about the status of the aircraft and its environment.

The Gannet has no such instruments but instead has a body covered with feathers.

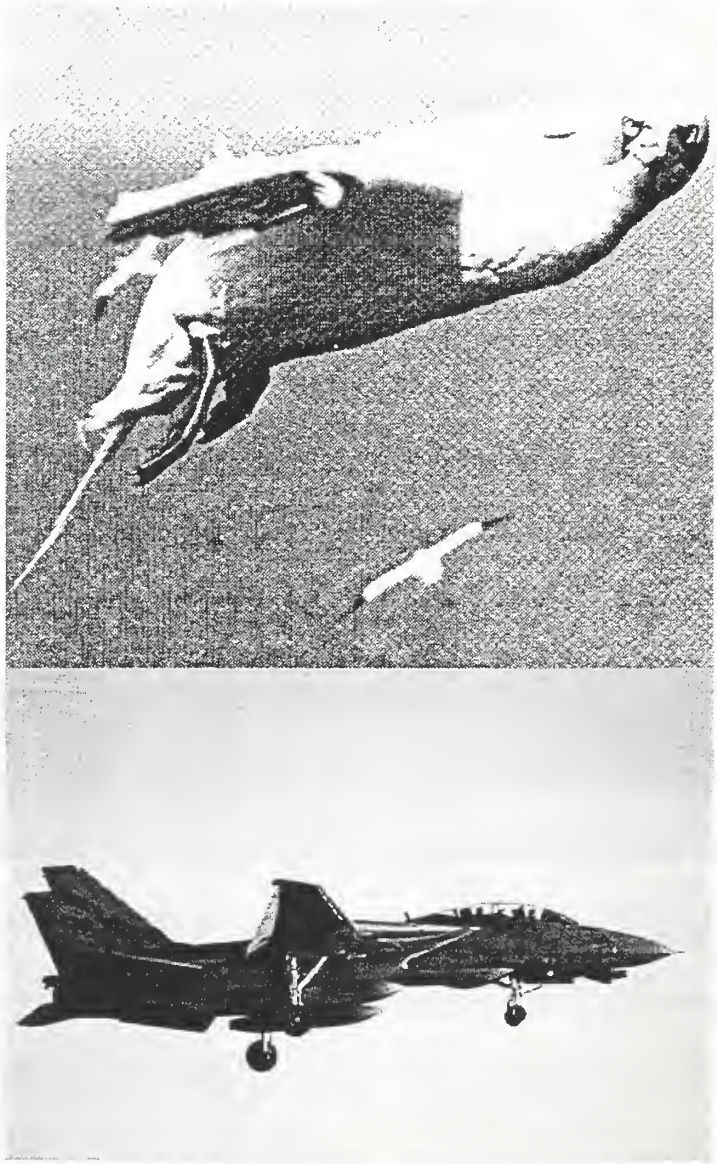


Figure 1. Gannet [Ref. 1] (top) and F-14 (bottom) in slow flight

Each feather ends in a nerve plexus that connects to the bird's nervous system and transmits continuous, real-time environmental information directly to its brain. Types of information

conveyed by this method include outside air temperatures, three-dimensional velocities, and air pressures at any point on the body. Additionally, the bird's skin-muscle-joint system transmits information to the Gannet's brain via the same nervous system about wing shape, wing and body position, and landing gear position.

A human pilot must take in all the same information presented on one and two-dimensional displays using only his eyes and ears, then process the information and apply it to a three-dimensional world. The Gannet, on the other hand, takes in three-dimensional information using the sense of touch, which can be easily applied to a three-dimensional world. The Gannet's eyes and ears are then used for the more important tasks of collision avoidance and target acquisition. Without the means to easily acquire and display the sorts of information enjoyed by the Gannet, human pilots are subject to potentially catastrophic conditions such as Spatial Disorientation caused by a loss of Situation Awareness.

## A. SITUATION AWARENESS AND SPATIAL DISORIENTATION

Spatial Disorientation (SD) is a triservice aviation problem that costs the Department of Defense more than $300 million annually in destroyed aircraft and is the primary cause of pilot-related mishaps in the Navy and the Air Force [Ref. 2]. In 1980-89, disorientation/vertigo was identified as the definite cause factor in mishaps that resulted in the loss of 38 lives and 32 aircraft [Ref. 3]. Of the 15 Navy aircraft lost to noncombatant action in the Desert Shield/Storm conflict, 7 were SD mishaps [Ref. 2]. Spatial Disorientation is a result of a loss of Situation Awareness (SA). Situation Awareness, as it relates to pilot orientation, has three components:

1. Spatial orientation of the operator and his platform relative to the earth.
2. Orientation relative to friendly and enemy forces.
3. Orientation relative to geography and the battlefield environment.

In aviation, the greatest threat to loss of SA is the loss of spatial orientation. The human body has three major sensory systems it uses for orientation [Ref. 4].

1. The visual system
2. The inner ear system of body balance
3. The skin-muscle-joint system

The combination of these three sensory systems provides the pilot a three-dimensional picture of the orientation of themselves and their aircraft. The visual system enables the pilot to distinguish colors, textures, and discriminate between different objects. In the aircraft, the visual system enables the pilot to see the critical "horizon line." The inner ear provides the pilot information on linear and rotational acceleration forces. The skin-muscle-joint system, using stretch receptors in the muscles, allows the pilot to sense acceleration forces and limb position.

While flying in visual daylight conditions, the pilot relies on his outside visual cues to provide correct orientation information to fly his aircraft. However, when the pilot loses his outside visual cues at night, in blowing sand, or during foul weather, he must rely solely on his flight instruments. Because of the acceleration forces unique to the flight environment, the inner ear system and the skin-muscle-joint system often provides false information. Many mishaps have shown that the erroneous influence of these systems combined with a few seconds of visual distraction is enough to cause catastrophic results.

## B. THE TACTILE SITUATION AWARENESS SYSTEM

As one solution to the SD problem, the Naval Aerospace Medical Research Laboratory (NAMRL) in Pensacola, Florida has developed a prototype system called the Tactile Situation Awareness System (TSAS). The primary objective of TSAS is to enhance pilot performance and reduce SD-related aircrew/aircraft losses and mission failures by simplifying the flight task. Providing pilots with a source of continuous and accurate information about the attitude and motion of their aircraft can reduce the effects of disorientation. The approach of TSAS is to provide continuous, non-visual information using the normally underutilized human sensory channel of touch. The sense of touch is intuitive, reliable, and not intrusive. The sense of touch is activated using vibrotactile stimulators to apply attitude and motion information taken from the aircraft's instruments to the pilot's torso. The vibrotactile sensation is similar to that of a silent pager, penetrating all environmental distractions while not interfering with the pilot's primary senses of sight and hearing. Additionally, tactile stimulation is a need-to-know technology. The pilot gets the information he needs when he needs it. The instruments tell the tactors, the tactors tell the pilot.

By providing continuous three-dimensional information to the pilot from the sensors on the aircraft, the SA of the pilot is increased through an improved man-machine interface. The Tactile Situation Awareness System at NAMRL consists of an array of electro-mechanical vibratory devices called "tactors" which are sewn into a vest under or inside of a flight suit, specialized hardware to control the tactor array, and a PC (Personal Computer).

Sensory information is input to the TSAS from a number of sources including an aircraft's attitude gyro and a computer-based simulation program. The TSAS receives the sensory input then processes and displays the resulting information on the three-dimensional array of tactors on the pilot's torso, Figure 2.

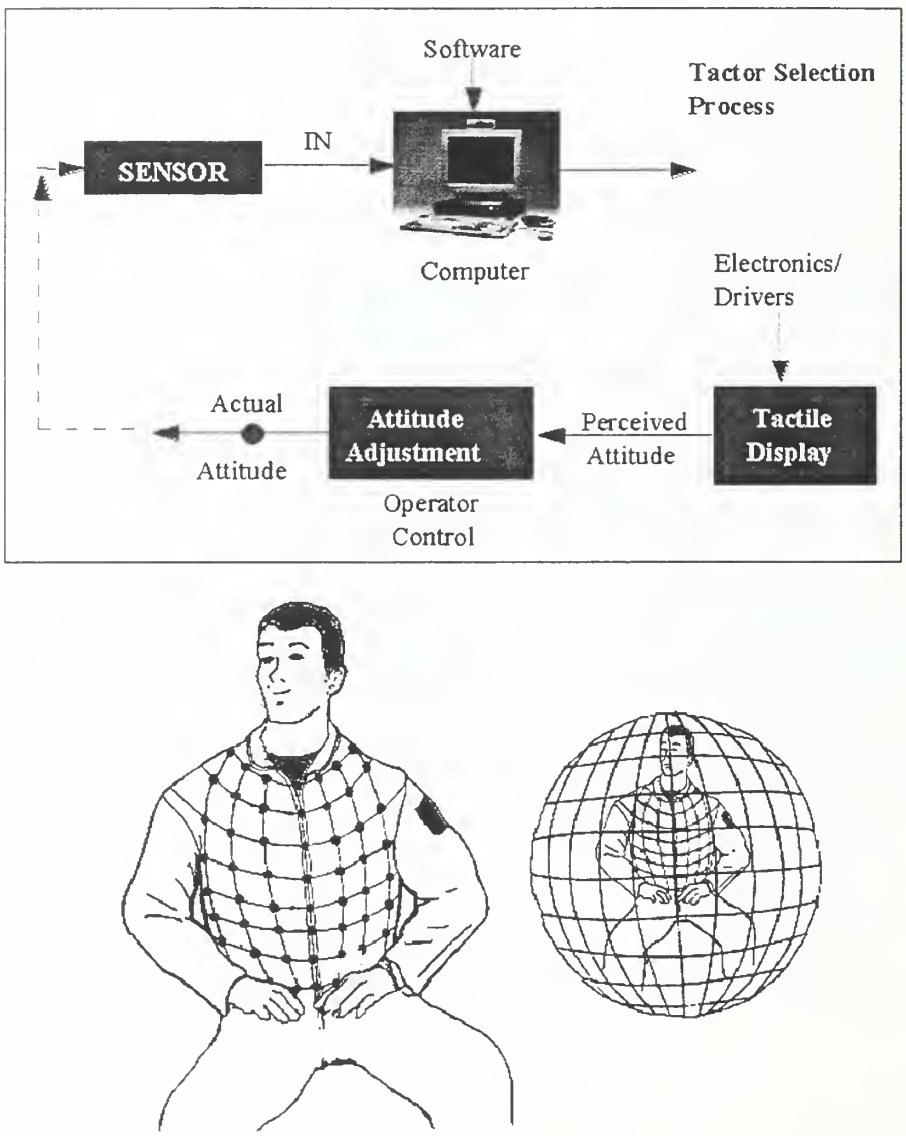



Figure 2. TSAS system configuration (top) and tactor display (bottom)

Different types of information may be displayed using electro-mechanical tactors by varying the position and frequency of the vibration on the pilot's body. Examples of the types of information displayed to a pilot include but are not limited to:

1. Orientation information
2. Navigation information
3. Threat or target information
4. A combination of orientation, navigation and threat information

The disadvantage of TSAS is that it is a research system and is not compatible with the crowded cockpit conditions of today's modern aircraft. This thesis presents the design of a miniaturized TSAS, which is more compatible with tactical environments.

## C. GOALS OF THE THESIS

This research provides the critical link that facilitates the transition from "technical to tactical" of the TSAS prototype. In particular, this thesis deals with the design of a microelectronic Tactor Interface Microcontroller System (TIMS), the primary purpose of which is to deliver tactile technology into the hands of the warfighter. Two principal objectives were established for this thesis. The first is to implement the Tactile Situation Awareness System in a palm-sized unit using commercial off-the-shelf (COTS) components. The second objective is to supplement TSAS with additional communication capability such that the system could interface with information sources such as a GPS (Global Positioning System) or the MIL-STD-1553 (Military Standard Specification 1553) Aircraft Internal Time Division Command/Response Multiplex Data Bus. A secondary

objective of this thesis is to present a system design which minimized the number of control wires connecting the TSAS controller to the tactor array.

## D. THESIS OUTLINE

The remainder of this thesis is organized as follows. Chapter II presents an overview of the Tactor Interface Microcontroller System and furnishes examples demonstrating how the technology can benefit the warfighter. Chapter III discusses in-depth the design and implementations of the Intelligent Tactor concept. Chapter IV discusses the design of the microcontroller system including CPU (Central Processing Unit), memory, power, and interface subsystems. Chapter V describes the process of converting the microcontroller system schematics from Chapter IV into a printed circuit board layout. Finally, Chapter VI contains testing results and suggestions for future work.

Four appendices are provided for reference: Appendix A contains the schematics of the Tactor Interface Chip logic, Appendix B includes the schematics of the Printed Circuit Board (PCB) design of the microcontroller system. Appendix C contains the PCB diagrams showing chip placement and etch trace routing, and Appendix D consists of a parts list with supplier information for the TIMS prototype printed circuit board.

This thesis is organized such that readers with different levels of technical knowledge can easily locate the relevant information. The reader who desires to understand tactile technology and its many applications need only read Chapters I, II, and VI to get an overview of the Tactile Situation Awareness System and the Tactor Interface Microcontroller System. The system-user or software programmer that needs to grasp the

operational complexities of TIMS must read Chapters I, II, III, VI, and the major sections of Chapter IV. The system-designer that desires to improve or reproduce the system design in hardware must read all chapters and appendices.

# II. THE TACTOR INTERFACE MICROCONTROLLER SYSTEM

This chapter presents an overview of the Tactor Interface Microcontroller System (TIMS). Research at the Naval Postgraduate School (NPS) in Monterey, California is focusing on miniaturization and optimization of the TSAS prototype. In order for TSAS to be used in the crowded cockpit conditions of today's military aircraft, it must be compact and power efficient. The system must also be able to interface with the computerized aircraft systems to obtain orientation and threat information.

The last section in this chapter furnishes examples demonstrating how the TSAS and TIMS technology can benefit the warfighter.

## A. SYSTEM REQUIREMENTS

The specifications for the TIMS prototype were formulated from discussions between researchers at NAMRL and the faculty and students at the NPS. What follows is a rough summary of the specifications [Ref. 3].

1. The system must be able to control a minimum of 40 tactors independently.
2. The system must activate the tactors using a user-defined voltage and frequency.
3. The system must be programmed in a high-level language such as C++ or ADA.
4. The size of the computer control system should be as small as possible, preferably palm-sized.
5. The system should be optimized for low-power operation and should be capable of running off of a battery or a plug-in DC power source.
6. The number of control lines connecting the tactor array to the computer control system must be minimized.
7. The system must be able to communicate with the MIL-STD-1553 aircraft bus.
8. The system must have standard RS-232/422 interfaces to communicate with other serial navigation devices and computers.
9. The system must not generate so much heat as to be uncomfortable to wear.

## B. SYSTEM CAPABILITIES

The Tactor Interface Microcontroller System, as designed, is logically divided into two sections:

1. The microcontroller system
2. The Intelligent Tactor

The microcontroller system consists of a powerful integrated CPU and communications controller, ten megabytes of memory, and a variety of I/O interfaces. The I/O interfaces include hardware to communicate with modern military and civilian aircraft, GPS receivers, host computers, and other devices utilizing standard serial interfaces. The microcontroller system is discussed in detail in Chapter IV.

The Intelligent Tactor is an electro-mechanical tactor attached to a Tactor Interface Chip, which controls the tactor's vibratory stimulus. The Intelligent Tactor concept provides for the distribution of processing workload over the entire system, relieving the CPU of micromanaging the entire tactor array. The Intelligent Tactor is discussed in Chapter III.

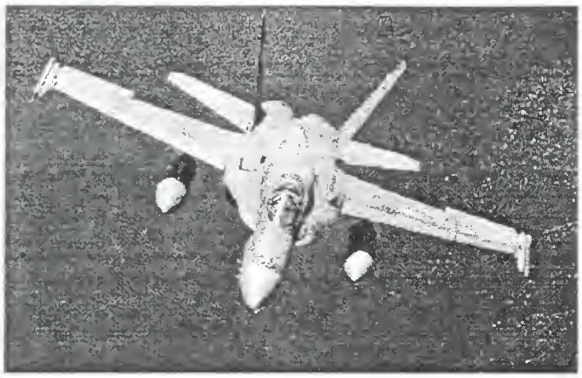
## C. APPLICATIONS

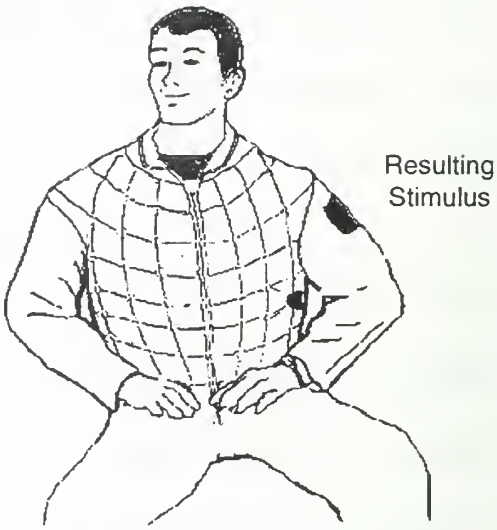
It is obvious that the TIMS project has applications in disorienting aviation environments where the pilot's primary senses are overloaded and there is a need for a three-dimensional display of information. TIMS can also be of great use in environments where visual and auditory communication is severely impaired. Since tactile communication is reliable, noiseless, and inherently non-visual it can be used in covert operations when noise

or light may pose a risk of detection. Just a few of the possible military applications of TIMS are spatial orientation awareness, flight simulator training, shallow water mine countermeasures, remotely piloted vehicles, land navigation and communication, and space applications.

### 1. Improving Spatial Orientation and Situation Awareness

Pilot disorientation is a major factor in pilot-factor aircraft mishaps in the U.S. Navy. The objective of projects like TSAS is to reduce mission failure, aircraft loss, and pilot loss due to pilot disorientation and to enhance pilot performance by simplifying the flight task. Disorientation can be reduced through improved pilot training and by giving continuous and accurate information to pilots on the attitude and motion of their aircraft. Currently, the only accurate sensory information available to pilots concerning their attitude and motion is visual interpretation of instruments or outside reference to the horizon. Most spatial disorientation mishaps occur when the pilot's visual attention is temporarily directed away from flight instruments to other tasks. The TIMS prototype would provide continuous information via the underutilized sensory channel of touch by applying attitude and motion information from the instruments to the torso of the body via vibrotactile stimulators. Normally in the aviation environment the sense of touch provides inaccurate information. Figure 3 presents an example TSAS prototype as an aid to spatial orientation.






TACTILE
INTERFACE

**30° ROLL LEFT**

Figure 3. TSAS system presenting orientation information to pilot

The T-34 TSAS flight demonstration project integrated a tactile display into a T-34 aircraft, Figure 4. A seven-event test operation was conducted to demonstrate that a pilot could maintain aircraft orientation using a tactile display during flight operations. Summary results showed that roll and pitch cues could be provided via a matrix array of vibrotactors

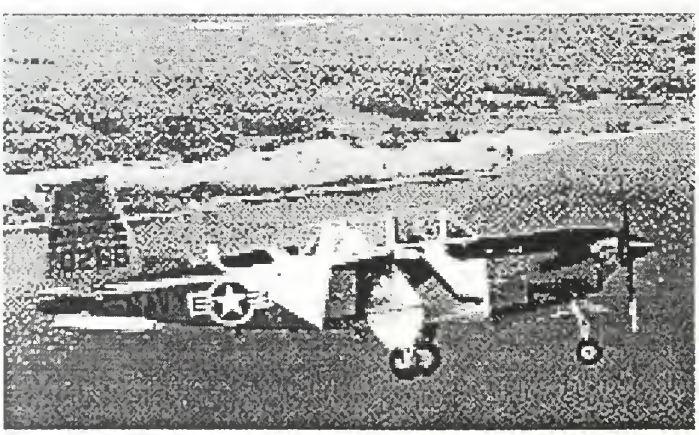


Figure 4. TSAS equipped T-34

incorporated into a vest under the flight suit. The test pilot in the rear seat was shrouded to block any outside visual cues and all flight instruments in the rear cockpit were removed. The test pilot flew the following maneuvers:

1. Straight and level for five minutes
2. Climbing and descending turns
3. Unusual attitude recovery
4. Loops and aileron rolls
5. Ground controlled approaches (GCA)

The test pilot successfully performed all maneuvers without visual cues, relying solely on tactile cues for attitude information. [Ref. 2]

The UH-60 TSAS flight demonstration project and Joint Strike Fighter (JSF) TSAS demonstration projects were follow-on efforts to the fixed-wing T-34 TSAS flight demonstration and integrated the tactile display into a UH-60 helicopter, Figure 5. Similar to the T-34 tests, TSAS pilots in the UH-60 performed flight test events with no visual cues.

Figure 5. TSAS equipped UH-60

The test pilots flew the following maneuvers: hover operations; straight and level; standard rate turns; unusual attitude recovery; and GCA. In the UH-60 helicopter hover flight tests, pilots using the tactor system in a visually impaired environment accurately detected drift velocity. Pilots testified that the hovering workload was significantly reduced in both Visual Meteorological Conditions (VMC) and Instrument Meteorological Conditions (IMC). One pilot was quoted as saying the tactors provided, "readily accessible and easily understood information that increases safety of flight." [Ref. 2]

### 2. Flight Simulator Training

During the initial phase of flight training, every new pilot and navigator spends many hours in flight simulators going over checklists and procedures, getting used to the cockpit environment, and learning to fly the aircraft solely using instruments. While the fledgling aviator is acclimating to the sights and sounds in the cockpit, his body is receiving none of the three-dimensional G-force accelerations that it will receive once airborne. The gravitational forces acting on the aviator's body in-flight stimulate his sense of touch and the muscle-joint system with false information and act as a distraction during the flight.

Integrating tactile communication into the flight simulator allows the pilot to get accustomed to accurate sensory input through the central nervous system before taking to the air. Once the pilot is accustomed to "feeling" the simulator, introducing new gravitational forces through the same sensory channel in-flight will have a reduced effect as a distraction. Introducing tactile communication starting at the simulator phase of flight training could reduce the amount of simulator training time required by facilitating the simulator-to-aircraft transition and would enhance pilot performance.

### 3. Unmanned Vehicle Operation

Due to the forces of acceleration acting on a pilot's body during flight, a pilot's tactile sense and muscle-joint systems transmit inaccurate attitude and orientation information via his central nervous system.

The pilot of a remotely piloted vehicle (RPV), Figure 6 or unmanned underwater vehicle (UUV) has a related but opposite problem. No tactile sensations, either true or

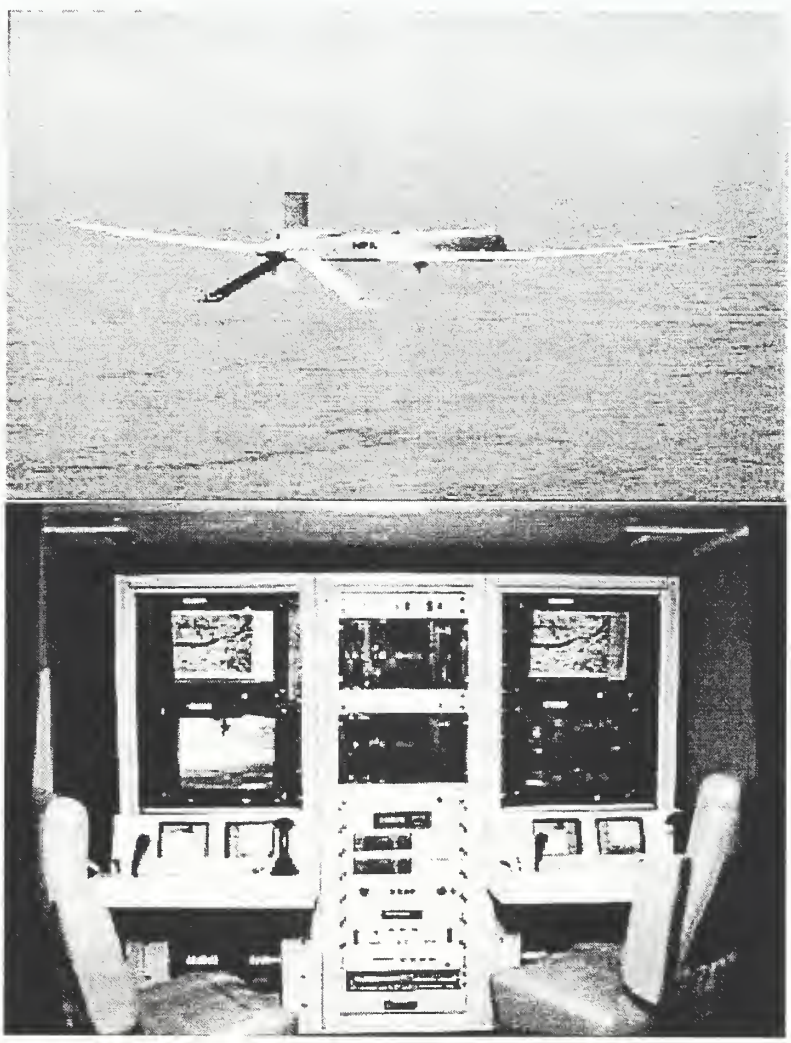


Figure 6. UAV and control station

false, are being transmitted via the central nervous system. Furthermore, similar to what occurs to a pilot in a manned aircraft, an RPV pilot's primary senses are overloaded with a variety of computer-generated displays, lights, and sounds necessary to control the RPV.

The use of tactile communication enables the RPV pilot to literally get a "feel" for orientation and velocity of the RPV. This increases situation awareness in a way not possible using standard one and two-dimensional displays. Additionally, the conveyance of information via the sense of touch partially relieves the load on the pilot's primary senses so the RPV pilot can concentrate less on the mechanics of flying and more on mission goals.

#### 4. Sea, Air, and Land Navigation

The application for tactile communication in the field of Very Shallow Water Mine Countermeasures (VSWMCM) is an extension of the TSAS project and is being researched in Panama City, Florida using the SEAL Inshore Navigation System (SINS) as the sensor source for TSAS [Ref. 5]. Navy SEAL (Sea, Air, and Land) divers wear suits of four tactors underneath their wetsuits. The computer system and tactor suit guides them along a pre-programmed mine-hunting path in up to 100 feet of water. Figure 7 is an image of a diver using the TSAS prototype in a simulated mine hunting exercise. Visibility is often poor in the VSWMCM environment. The use of the TSAS better enables the divers to look for mines and to perform tasks other than navigation.

In July 1995, SINS/TSAS feasibility tests were conducted at Naval Amphibious Base, Coronado, CA, with members of the VSWMCM Test Detachment. A total of six dives were completed. Twelve laps around a triangular course represented a total of

Figure 7. Diver using TSAS prototype for mine hunting

approximately 6144 yards and total bottom time for all dives was approximately six hours.

Some conclusions from the computer data and diver evaluations follow [Ref. 5]:

1. Tactile actuators used in an underwater navigation environment are a feasible alternative or enhancement to visual displays.
2. Underwater navigation cross-track error (deviation) from the baseline navigation course was insignificant for both the SINS and SINS/TSAS tests.
3. A majority of the divers felt that tactor technology was easier to use, provided better navigation, and preferred the tactile technology over visual only displays.
4. All of the divers felt that navigation operational capabilities could be enhanced with tactile technology.

After touting the advantages of tactile communication use in the air and underwater, it is not surprising to imagine uses for TIMS prototype for Special Forces on land as well.

Because of the following advantages, tactile communication can be extremely useful in land navigation and covert communications.

1. Low signature
2. Analogous to reality
3. Good representation of 3-D space
4. Monitoring/attention of user not required
5. Utilizes an otherwise unused sense
6. Provides backup to other senses
7. Used to alert or call attention
8. Reduces information overload

Global Positioning System receivers have been widely used for land navigation for several years. Most of the hand-held GPS devices are simple to operate and provide accurate positional readings all over the world. One drawback to using a GPS receiver is they are hand-held and require the user's visual attention. When climbing, hiking rough terrain, or carrying a lot of equipment it is impractical if not impossible to hold and constantly monitor any portable electronic device. The TIMS prototype includes an I/O interface for GPS receivers that will enable the user to receive tactile navigational cues from the GPS receiver without having the receiver in hand or within sight.

Although the phrase "tactile communication" appears several times in this introduction, it has always referred to a man-machine interface. However, there is also an application for the TIMS prototype in man to man communication. The communication that tactors provide to the wearer of the system is silent and non-visual, two important requirements for covert operations. Using a wireless communications interface, information can be sent to wearable TIMS prototypes on several squad members in a patrol situation. The squad leader, using a palm-sized touchpad, can send such commands as,

stop, crouch, go left, and go right. However, the choice of wireless communication method used to transmit information between squad members is subject to detection by hostile forces and must be considered carefully in the design of the system.

Currently this same communication can and does take place via hand signals, Figure 8, but hand signals are only effective when the receiver is looking at the sender and with

Figure 8. Land navigation hand signals

sufficient lighting. Man-to-man tactor communication presents an alternative form of silent covert communication that may be used in all lighting conditions.

### 5. Space Applications

Because tactile technology operates in so many different environments and in so many modes, it has as many applications in space as on earth, Figure 9. Since there is no formal "down" in space, tactors can provide the astronaut with a reference point or anchor inside or outside a space vehicle. Additionally, tactors can improve Situation Awareness and navigational ability during extravehicular activity (EVA).

A tactile display that was proposed to increase an astronaut's situational awareness during an extravehicular activity has been developed and tested [Ref. 6]. The Tactor Locator System (TLS) is a non-intrusive, intuitive display that can be configured to convey

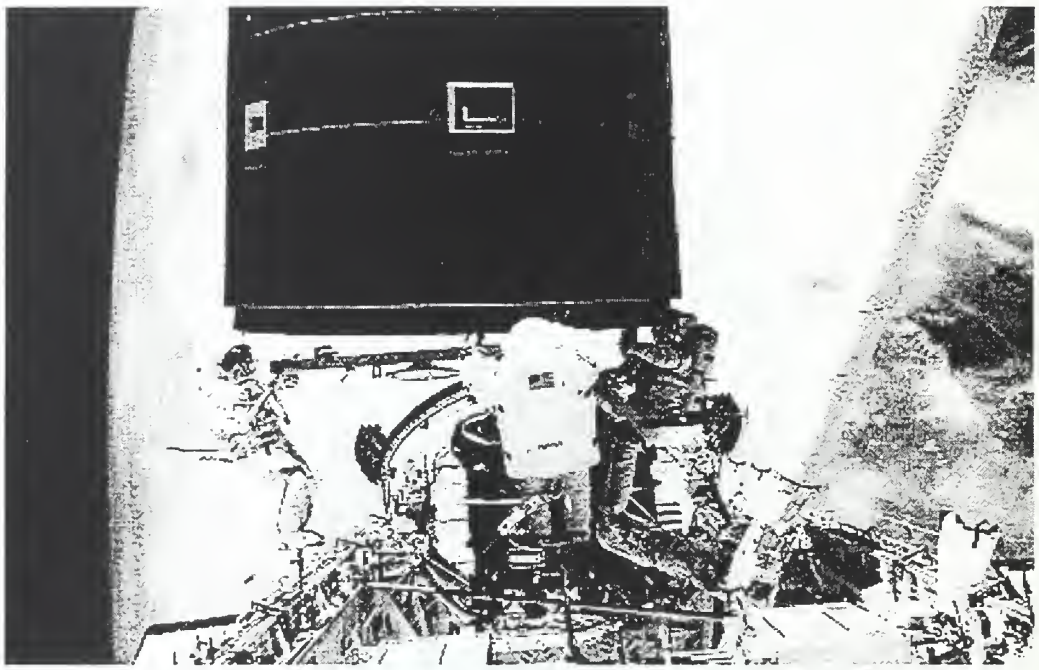


Figure 9. EVAs use TIMS to ensure situation awareness in space

position, velocity and orientation information via a vibrotactile stimulus applied to the torso region. With the construction of the International Space Station, the number of EVAs required of the astronauts will increase by a factor of five over those currently conducted for the Space Shuttle program. The tasks to be performed will range from the piece by piece construction of the station, to its maintenance once completed.

For any EVA, it is important that the astronaut maintains a high a level of situational awareness. Exposure to weightlessness can lead to conflicting sensory cues, resulting in decreased situational awareness. Situational awareness is aided during space shuttle EVAs

because most are conducted in or near the cargo bay or air lock in full view of other crewmembers. However, space station EVAs will most likely occur considerable distance from such reference points or other crewmembers. Therefore, astronauts conducting space station EVAs are in need of an additional aid to assist in navigation, tracking and orientation. The TLS was designed to provide cues to complement the visual system during an EVA. Obviously, the TIMS has applications in miniaturization and optimization of the TLS prototype as it does for the TSAS prototype.

# III. THE INTELLIGENT TACTOR

The Intelligent Tactor is the cornerstone of the TIMS project and hence is presented first in this thesis. The Intelligent Tactor concept provides for the distribution of processing workload over the entire system. Simple, highly repetitive tasks are processed and executed by the TIC (Tactor Interface Chip) attached to each tactor rather than by the microcontroller as in the current TSAS prototype. In the TIMS, the microcontroller communicates to the TIC via a digital high-speed serial communication line. This relieves the CPU of the cycle-greedy chore of micromanaging the tactor array and enables the CPU to focus its processing power on sensor information collection and analysis. The combination of the TIC and the electro-mechanical tactor is called the Intelligent Tactor. This chapter presents the Intelligent Tactor concept and proposes a TIC design and communications protocol.

## A. INTELLIGENT TACTOR SPECIFICATION

The goal of the Intelligent Tactor is to transfer as much functionality as possible from the computer to the tactor while keeping the tactors small enough and at a low enough temperature to be worn comfortably against the skin under a flight suit. The tactor operation specifications [Ref. 3] follow.

1. The system must be able to control a minimum of 40 tactors.
2. When vibrating, the tactor is driven at a constant voltage modulated on and off at a frequency of 250 Hz called the Carrier Frequency (Figure 10).
3. The Carrier Frequency waveform is modulated on and off at the Modulating Frequency.
4. The period of the Modulating Frequency ranges from 0-4000 milliseconds (Repetition Period).

5. The amount of time the tactor may be vibrating during one period (Excitation Interval) ranges from 0-1000 milliseconds.
6. Changing the on/off state of all of the tactors simultaneously must be accomplished in less than one millisecond.
7. The system must have a real-time operating system

The Intelligent Tactor must be able to conform to the above specifications using only the commands and signals sent from the TIMS microcontroller. The process of designing an Intelligent Tactor is presented in three parts: Defining the method of

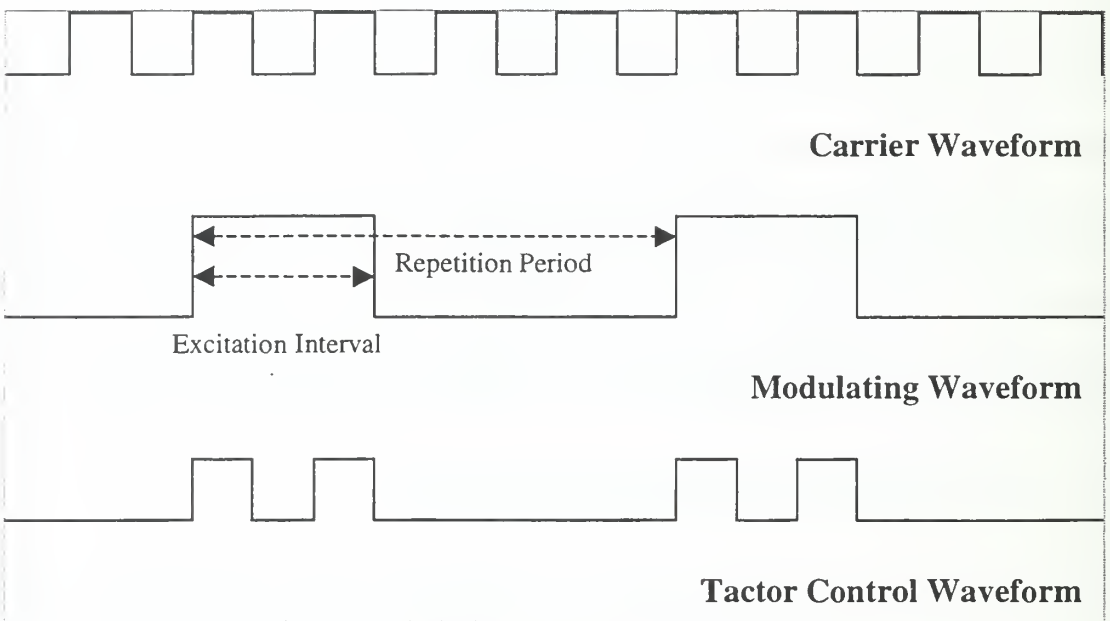



Figure 10. Tactor control waveform generation

communication between the Intelligent Tactor and the TIMS microcontroller, defining a command set to be used over the communications link, and designing the Intelligent Tactor logic to implement the command set. The TIC logic and command set presented in this chapter is only one of many possible designs. Furthermore, advanced TIC and command set features are implemented in this design that are not necessary for simple tactile communication operation. The advanced features serve to demonstrate the potential of the

Intelligent Tactor concept. Therefore, initial hardware implementations of this design may only implement a subset of the logic and command set features.

## B. MICROCONTROLLER TO TACTOR COMMUNICATION

One of the primary objectives of TIMS is to minimize the number of physical control lines connecting the microcontroller to the tactor suit. In the current TSAS configuration, each tactor in the system is connected to one control wire and a common ground wire. Modulating the voltage on the control wire turns the tactor on and off. Consequently, a suit of 40 tactors performing only the most basic functions requires 41 control wires. A bundle of 41 or more wires is a thick and unwieldy piece of equipment in the cramped cockpit environment of today's modern aircraft. Additionally, each connector on each of the wires is a potential point of failure in the system.

Under the Intelligent Tactor concept, control of each tactor is being executed by the TIC located at the tactor itself. Consequently, the communication between the TIC and the microcontroller will be much less frequent than the computer-tactor communication in the TSAS prototype. TIMS takes advantage of this relative infrequency of communication by implementing a set of standard commands for communication between the microcontroller and the TIC. The commands transmitted to the tactors are multiplexed onto a single control wire connecting all Intelligent Tactors in the tactor array. Using this method, the number of wires required to control an array of 40 Intelligent Tactors can be reduced to a mere handful of control lines. An example TIC is presented later in this chapter utilizing only seven control lines.

The following section is a review of serial communication with a particular emphasis on UART (Universal Asynchronous Receiver/Transmitter) and USART (Synchronous UART). This section may be skipped if the reader is already familiar with these concepts.

### 1. Review of Synchronous UART

Serial communication normally consists of transmitting binary data across an electrical or optical link such as RS232 or V.35. The data, being binary, is usually represented by two physical states. For example, in RS232, a positive 12 volts represents a '1' and a negative 12 volts represents a '0.' The accurate decoding of the data at the remote end of the link is dependent on the sender and receiver maintaining synchronization during decoding. The receiver must sample the signal in phase with the sender. If the sender and receiver were both supplied by exactly the same clock source, then transmission could take place forever with the assurance that signal sampling at the receiver was always in perfect synchronization with the transmitter. This is seldom the case, therefore in practice the receiver is clocked locally and is periodically brought into synchronization with the transmitter. It is left to the internal clocking accuracy of the transmitter and receiver to maintain sampling integrity between synchronization pulses.

Asynchronous communication is the method most widely used for simple terminal serial communications. In asynchronous serial communication, the electrical interface is maintained at the "idle" state. The start of transmission of a character is indicated by a change in the signal level. At this point, the receiver starts its clock. After one bit time, called the start bit, 8 bits of true data are sent, followed by one or more stop bits. The

receiver tries to sample the signal in the middle of each bit time. The byte will be read correctly if the line is still in the intended state when the last stop bit is read. Thus, the transmitter and receiver only have to have approximately the same clock rate, rather than exactly the same clock rate. A little arithmetic will show that for a 10 bit sequence, the last bit will be interpreted correctly even if the sender and receiver clocks differ by as much as five percent, Figure 11.

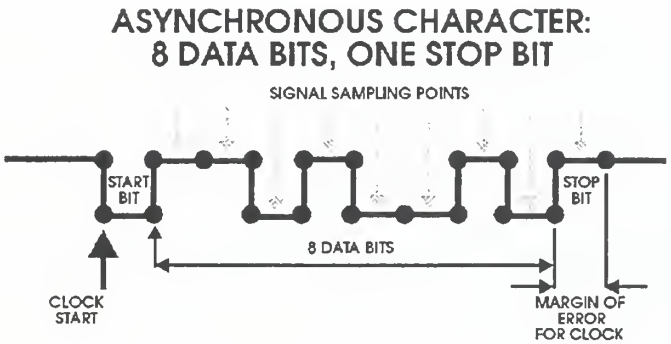



Figure 11. Asynchronous communication frame

Asynchronous communication is relatively simple and therefore inexpensive. However, it has a high overhead, in that each byte carries at least two extra bits: a 25% loss of line bandwidth. A 56kbps line can only carry 5600 bytes/second asynchronously, in ideal conditions.

Synchronous communication is usually much more efficient in use of bandwidth than asynchronous. The data field is usually large in comparison to the flag, control, address, and CRC (Cyclic Redundancy Check) fields, therefore there is very little overhead. A 56kbps synchronous line can be expected to carry close to 7000 bytes per second (i.e., 56000/8, whereas the asynchronous data rate would be 56000/10). Another advantage of

synchronous communications is the frame structure allows for easy handling of control information. There is a natural position (usually at the start of the frame) for any special codes that are needed by the communication protocol, Figure 12. Examples of synchronous


Figure 12. Synchronous serial communication frame

frames are HDLC (High-Level Data Link Control) and ATM (Asynchronous Transfer Mode).

Synchronous UART or USART uses the format of asynchronous communications but sends a clock signal along a separate line to synchronize the data at the receiver. There is still a bandwidth loss due to the start and stop bits. However, the two features that make USART ideal are the speeds at which it can be transmitted are much greater than UART, and most computers have the capability to decode UART format. At present, there is no synchronous frame standard common to most computers. Figure 13 summarizes the synchronous and asynchronous protocol formats.

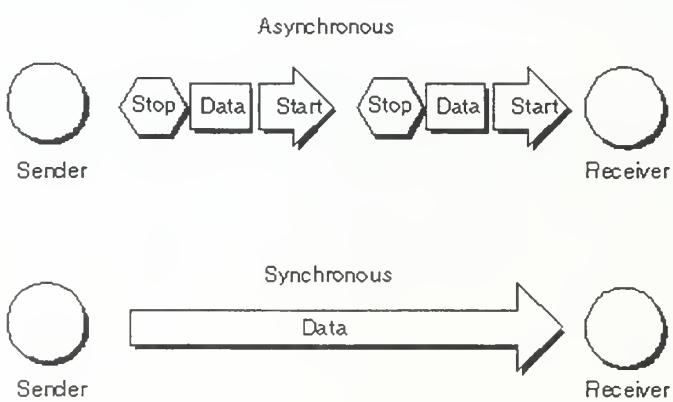



Figure 13. Synchronous and asynchronous communication

### 2. Serial Communication Command Set

The serial communication method for the TIC is USART with one start bit, one stop bit, an odd parity bit, and eight data bits for every byte of data. Since all TICs are connected to the microcontroller via a single serial data line, some method of addressing must be implemented such that each TIC responds to a command sent to it and does not respond to commands sent to other TICs. Consequently, a three-byte command word structure was designed to communicate basic functions to the TIC.

The first byte of the command word contains the binary address of the TIC. The first bit of the address is always a one, indicating the byte is an address byte and not a data byte. The seven remaining bits constitute the binary address of the TIC. The all-ones address is reserved as a broadcast address, which is used to send all tactors in the system a command simultaneously. The remainder of the addresses, 0000000 to 1111110, constitutes 127 different addresses, each of which may be assigned to a TIC. This more than meets the minimum 40-tactor specification.

The second byte of the three-byte command word determines the length of the Excitation Interval (EI). The EI is the amount of time the tactor is on (i.e. when it is vibrating) and is a component of the Modulating Waveform, Figure 10. The first two bits of the second byte are always 00, which identify the byte uniquely as a data byte. This leaves six bits to represent the EI. The all zeros combination is a special purpose command that tells the TIC to turn the tactor off. The remainder of the binary combinations, 000001 to 111111, equates to the decimal values 1 to 63. Since the specifications for the EI call for an interval of 0-1000 milliseconds, simple calculations reveal that multiplying the decimal

values of the EI by an interval of 16 milliseconds results in an EI range of 16-1008 milliseconds. An EI of zero milliseconds is, of course, the same as turning the tactor off. It turns out that the resulting EI range is well within timing specifications, given the sensitivity of the human skin.

The third byte of the three-byte command word determines the length of the Repetition Period (RP). The RP is the amount of time it takes for the tactor to complete one on/off cycle. Referring back to Figure 10, the EI is the on-time of the tactor, and the RP is the EI plus the off-time of the tactor. The first two bits of the third byte are always 00, which leaves six bits to represent the RP. The all zeros combination is a special purpose command that tells the TIC to turn the tactor on continuously. The remainder of the binary combinations, 000001 to 111111, equates to the decimal values 1 to 63. Since the specifications for the RP call for an interval of 0-4000 milliseconds, simple calculations reveal that multiplying the decimal RP by an interval of 64 milliseconds results in an RP range of 64-4032 milliseconds. Again, it turns out that the resulting RP range is well within timing specifications given the sensitivity of the human skin.

When the entire three-byte command word is received by the TIC, the command word is checked for a correct address and structure. The addressed TIC executes the command and modulates the tactor on and off, repeating the EI/RP cycle designated in the command word until the TIC receives another command.

Although the majority of the commands sent to the TICs are EI/RP cycles, there are many desirable auxiliary functions that the TIC may perform. Thus, a set of standard auxiliary commands is implemented along with the EI/RP commands.

The first byte of the Auxiliary Command (AC) word is the address byte as described earlier in this section. The second byte of the auxiliary command word is the auxiliary command, instructing the tactor to perform a single operation. The execution of the auxiliary command may or may not interfere with the current EI/RP cycle of the tactor. The first two bits of the auxiliary command byte are always 01, which distinguish an AC from an address, EI, or RP byte. The next two bits further break the auxiliary command into four sub-groups. If the second two bits of the auxiliary command byte are 00, the remaining four bits determine the Carrier Frequency driving the tactor. If the second two bits are 01, the remaining four bits determine the voltage level supplied to the tactor. These two command functions are unimplemented in this design. The remaining auxiliary command combinations are reserved for single commands and future TIC expansion.

Only two auxiliary commands have been implemented in this design. One enables the TIC to perform a system reset, returning the TIC to a known initial state. The other enables the TIC to transmit serial data back to the microcontroller. These functions will be discussed in detail in the logic design description of the TIC. Figure 14 summarizes the three-byte and auxiliary two-byte command word structures. As noted earlier in this chapter, the command set in Figure 14 represents one possible command set. Furthermore, the command set includes advanced commands not necessary for basic TIC operation. Reducing the complexity of the command set also reduces the complexity of the logic which implements the command set.

| Byte 1 | Bit Pattern | Byte 2 | Bit Pattern | Byte 3 | Bit Pattern |
|---|---|---|---|---|---|
| **TIC Address** | $1a_6a_5a_4a_3a_2a_1a_0$ | **Excitation Interval** | $00e_5e_4e_3e_2e_1e_0$ | **Repetition Period** | $00p_5p_4p_3p_2p_1p_0$ |
| *Broadcast* | *11111111* | *Constant Off* | *00000000* | *Constant On* | *00000000* |
| **TIC Address** | $1a_6a_5a_4a_3a_2a_1a_0$ | **Aux Commands** | $01c_5c_4c_3c_2c_1c_0$ | | |
| *Broadcast* | *11111111* | *Send Feedback* | *01100001* | | |
| | | *Reset TIC* | *01100000* | | |
| | | Variable Carrier Wave Frequency | $0100f_3f_2f_1f_0$ | | |
| | | Variable Tactor Control Voltage | $0101v_3v_2v_1v_0$ | | |
| | | Single Commands | $0110x_3x_2x_1x_0$ | | |
| | | Unused | $0111x_3x_2x_1x_0$ | | |

Note: *Constant On* and *Constant Off* commands override all other EI/RP commands. If both are sent at the same time, the *Off* command overrides the *On* command.

Figure 14. Command set summary for two and three-byte commands

An additional feature has been designed into the TIC logic to enable the microcontroller to conserve bandwidth while sending the same commands to multiple tactors. When the address byte of a two or three-byte command word is sent, the TIC corresponding to the address begins waiting for an EI or auxiliary command byte. A tactor that receives a valid address and is waiting for a command byte is said to be in the "active" state. In the active state, a TIC ignores all address bytes until it receives a command byte. Multiple tactors may be placed into the active state before the command byte(s) are sent. For example, to execute a specific EI/RP cycle on 20 tactors using three-byte command words, it would take 20×33=660 bits including overhead bits. The same result can be

obtained by activating 20 tactors and then sending the two EI/RP command words. This requires only (20×11)+(2×11)=242 bits. The result is a 60% savings in bandwidth.

### 3. Intelligent Tactor External Signals Interface

The TIC example presented in this document requires seven signals to implement all TIC functions. The list below summarizes the required input signals.

1. Serial Data Line
2. USART Clock Line
3. 250 Hz Carrier Frequency (Tactor Power)
4. Tactor Ground
5. Chip Power
6. Chip Ground
7. Reset Signal Line

The first two are the serial data line and the USART clock line. All communication from the microcontroller to the TIC takes place on these two signal lines. It is possible to combine the two signals by embedding the USART clock signal in the data on the serial line using Manchester encoding. At this time, it is simpler to add one signal line than to modify the TIC logic to decode an embedded clock signal. This modification is an exercise left to future designers.

The third signal is a 250 Hz 50% duty cycle square wave Carrier Frequency signal that drives the tactor. This signal is modulated by each TIC individually. Recent research [Ref. 10] has suggested the USART clock and the 250 Hz clock should be combined into one signal. Under this scheme, the 250 Hz clock signal would be generated at the TIC using the high-speed USART clock signal, thereby eliminating one more signal line. Designers are strongly cautioned when combining these two signals due to the resulting loss of TIC flexibility. It is important that the USART clock and the clock that supplies the

tactor signal be independent. The independence allows the TIC to communicate with high-speed devices (i.e., MPC860 based system) and low-speed devices (i.e., laptop computers) without the use of additional interface hardware. Furthermore, it may not be possible for the prototype TICs to provide enough power to drive the more current-hungry tactors, thus having a separate 250 Hz line supply the current that the USART line cannot is very advantageous. Furthermore, while it is advantageous to transfer simple, repetitive processing tasks to the TIC, clock generation does not fall into this category. The microcontroller system presented later in this document is specifically designed to be ideal for such tasks. Therefore, this author recommends that while the system is still in the prototype phase, it is crucial to keep the clocks separate to maintain system flexibility for testing and evaluation.

Signals four, five, and six are power and ground for the TIC and the tactor. The seventh signal, a reset line, is not critical to TIC operation but it is important for system safety and integrity. The reset line should be asserted at system startup to initialize all TICs to a known state. Additionally, it should be used as a safety measure such that asserting the line shuts down all tactors in the case of system failure.

The TIC has only two outputs. The first is the tactor control signal. This signal connects directly to the electro-mechanical tactor and causes the device to vibrate on and off at the desired frequency. The second signal is the serial data feedback line. This serial bus line connects to each TIC similar to the input serial data line. At this point, there is no hardware in place to ensure two TICs do not transmit on the feedback line at the same time. All control over the feedback transmissions is performed in software. It may be necessary

in future designs to add another line connected to each of the tactors to act as a transmission bus arbitrator. The line would perform the wire-OR function and any TIC desiring to transmit on the feedback line would first check the arbitration line. If the line were pulled low by a TIC, then no other TIC would transmit. If the line were high, then the TIC requesting to transmit would pull the line low and proceed to transmit. See Reference 7 for details.

## C. TACTOR INTERFACE CHIP LOGIC DESIGN

The presentation of the TIC logic design in this section is from a top-down approach. To read this section from beginning to end requires an understanding of the function of a few basic logic building blocks. This section assumes knowledge and functional understanding of the seven basic logic gates, T and D edge-triggered flip-flops, simple multiplexers, synchronous up and down counters, latches, and shift registers.

The most abstract view of the TIC has already been presented in the previous sections but will be reiterated here in Figure 15 for completeness. All signals are present in the logic design with the exception of the power and ground connections. Power and ground connections typically do not appear in the logic design but instead belong in the chip-level implementation of the logic design.

One level of abstraction down from the TIC itself are the seven main modules that comprise the principal logical functions of the TIC. Each module has a specific function and a fixed set of inputs and outputs.

The seven main modules, listed in a general order of data flow, are the Serial Data Shift Register, Serial Data Protocol Check, Data Decoder, Command Processor, Auxiliary Command Processor, Serial Data Feedback Module, and Tactor Driver Module.

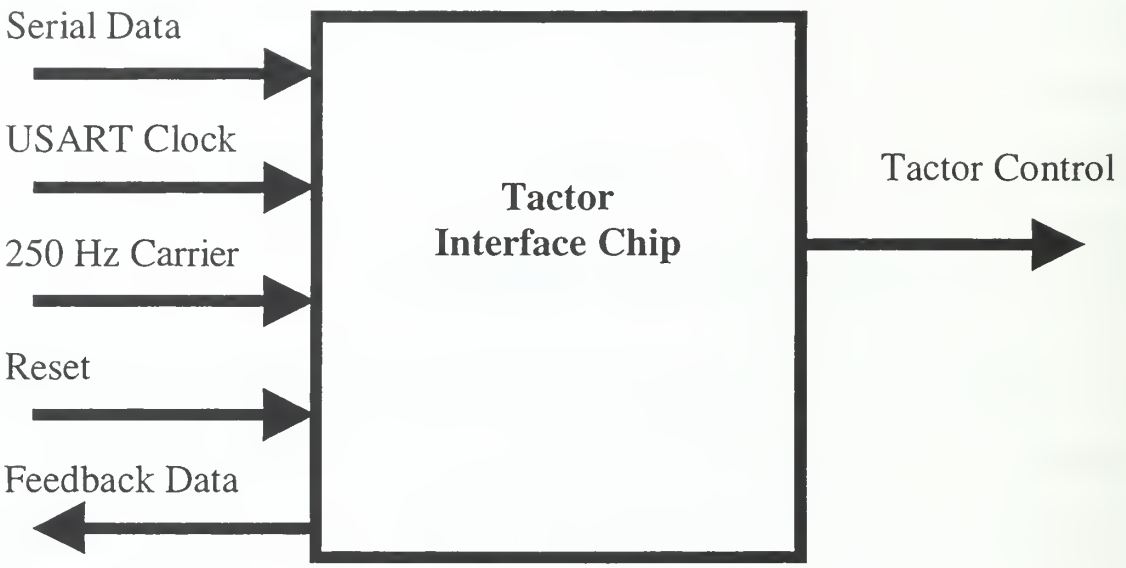



Figure 15. TIC diagram

### 1. Seven Principal TIC Modules

The functions and interfaces of the seven principal modules are summarized in Figure 16. The seven principal modules can be grouped together to form five basic TIC tasks.

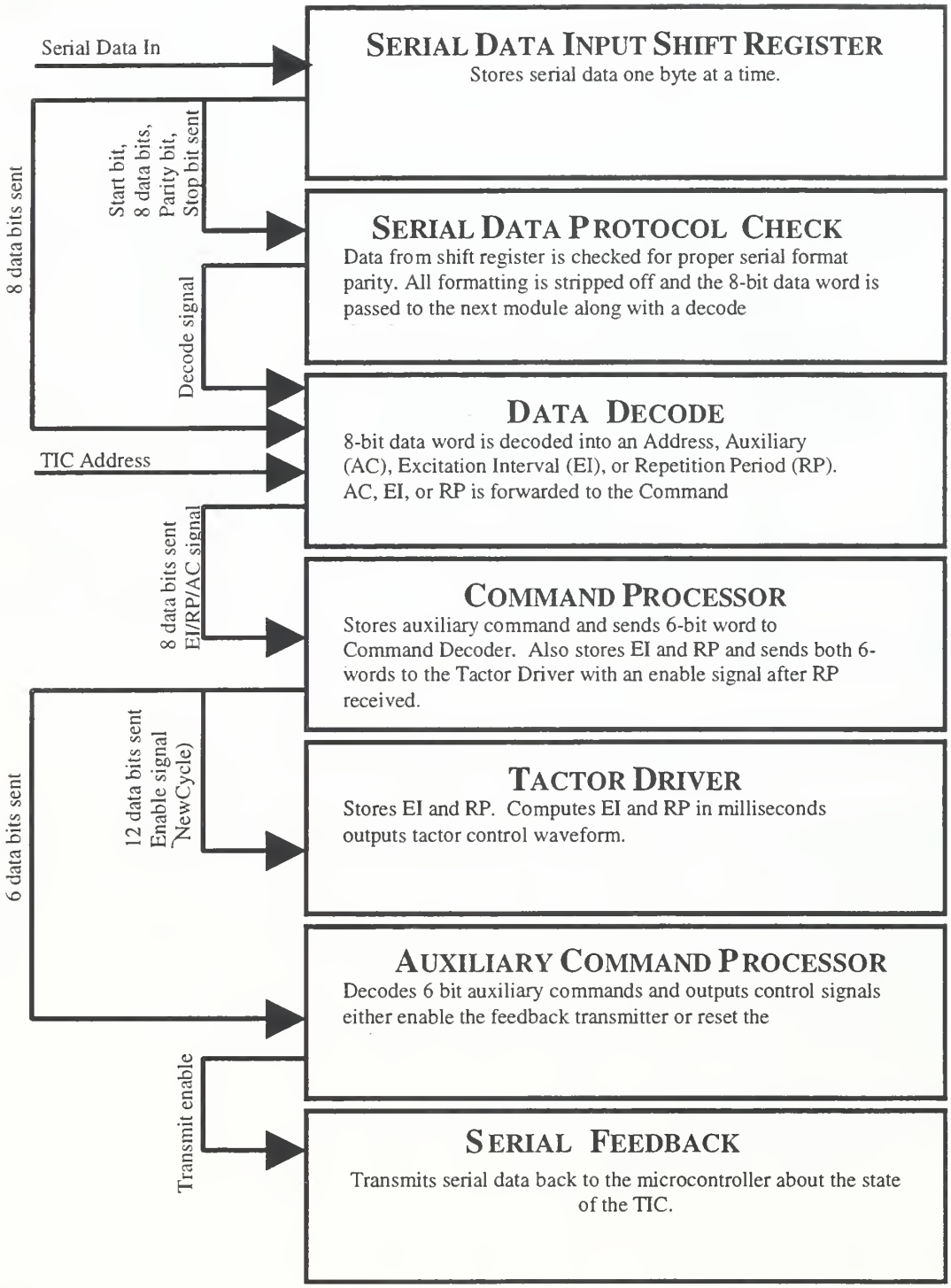



Figure 16. Descriptions and interfaces of the seven principal modules

The first two modules that manipulate the input serial data are the Serial Data Shift Register and the Serial Data Protocol Check Module. Figure 17 shows the two modules and their relationship to each other. Together, they perform the task of storing the serial data as it arrives from the microprocessor and checking the serial data for proper serial

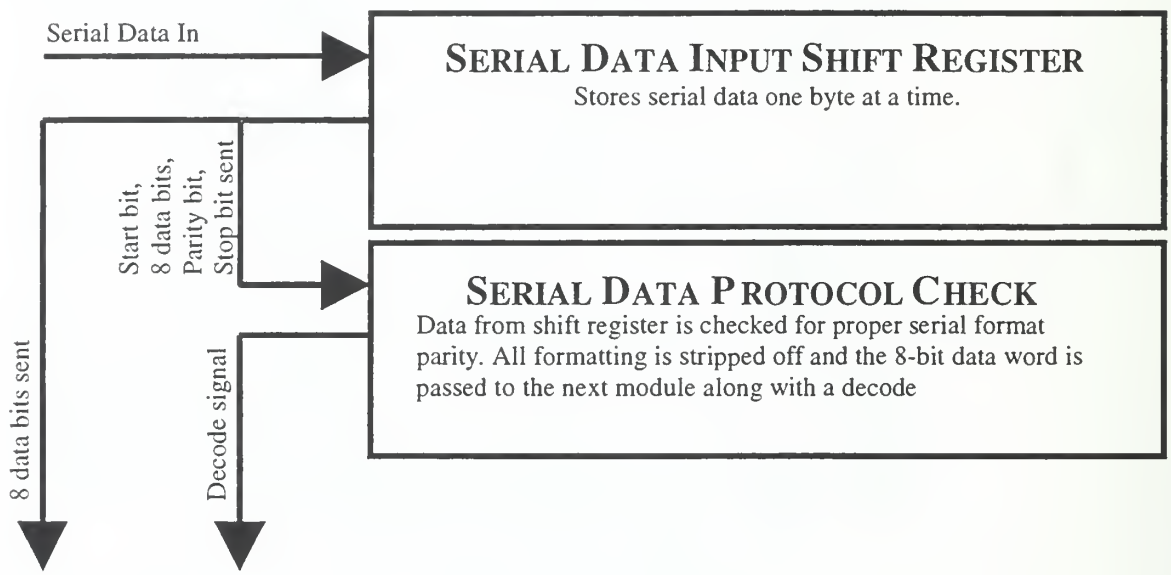



Figure 17. Serial Communication Protocol Check module

communication format. If the format is found to be correct, the overhead bits are stripped off and the eight data bits are sent in parallel to the Data Decode Module for storage and processing. A decode signal is also sent to the Command Decode Module to indicate that an entire byte has arrived and is ready for processing. If the format is found to be incorrect, an error signal is generated.

The Data Decode module, shown in Figure 18, is the heart of the communications portion of the TIC. The Data Decode module receives the eight data bits and decode signal from the previous module. The eight data bits may be decoded into an address, EI, RP, or

AC. If an address is decoded that matches the TIC address, the TIC is placed in the active state. If the TIC is already in the active state and decodes a valid command, the eight data bits and a signal indicating whether the command is an EI, RP, or an AC is sent to the Command Processor modules.

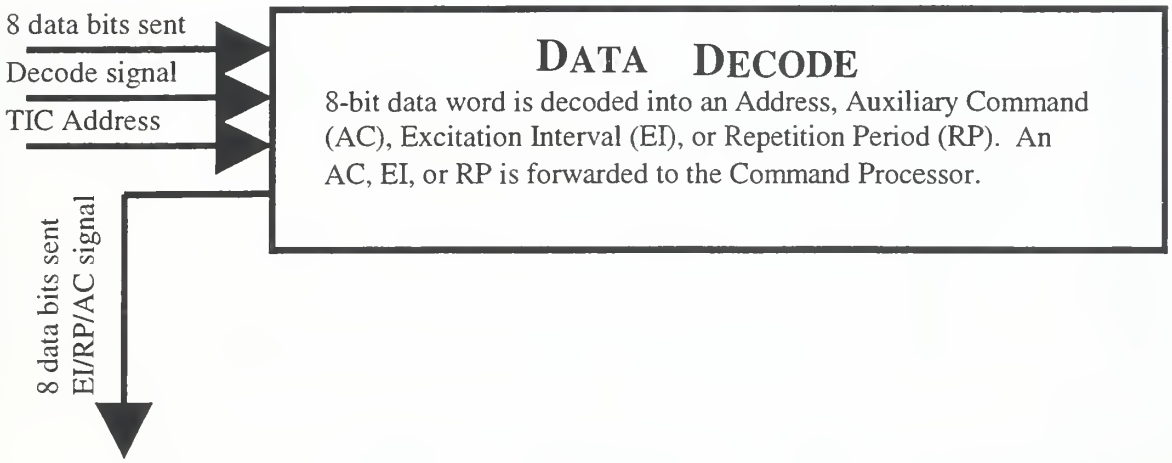



Figure 18. Data Decode module

The Command Processor modules include the Command Processor and the Auxiliary Command Processor as shown in Figure 19. When the Command Processor module receives EI data, it stores the bits and waits for RP data. When the RP data is received, both the EI and the RP are sent to the Tactor Driver along with a signal enabling the Tactor Driver to begin a new cycle using the new EI/RP data. If the Command Processor receives an AC, it stores the data and the Auxiliary Command Processor module decodes the command. If the Auxiliary Command Processor correctly decodes one of the two commands it recognizes, it will either send a reset signal to the TIC or enable the Feedback Transmitter module.

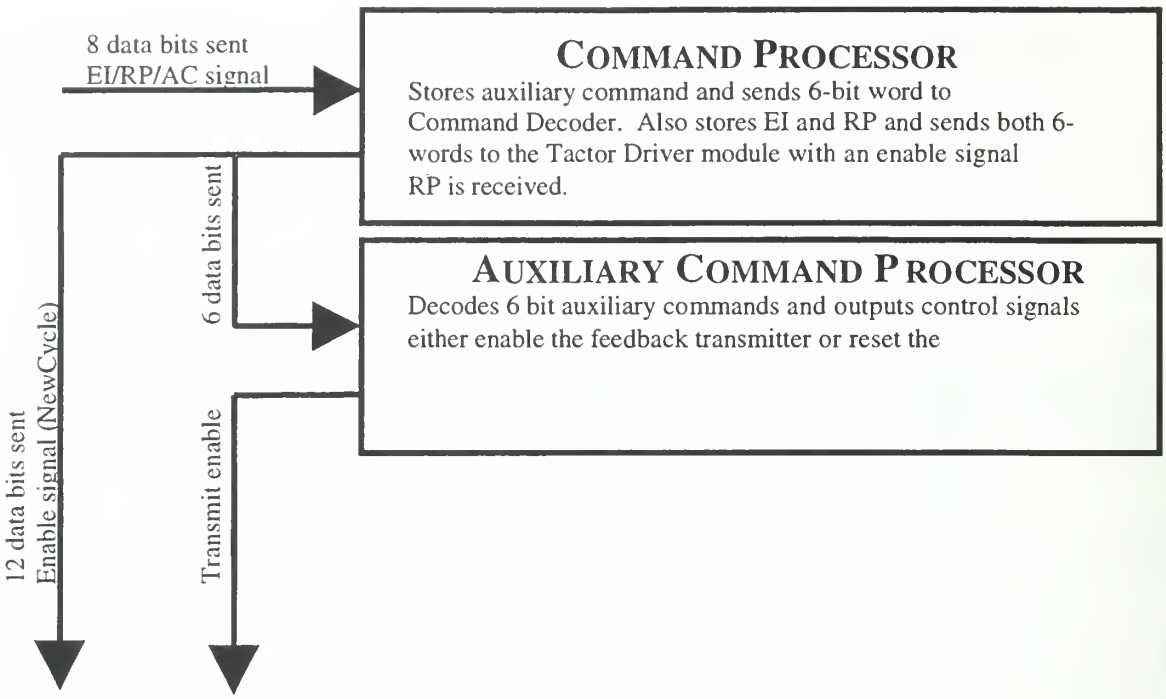



Figure 19. Command Processor modules

The last two modules, Figure 20, are grouped together not because they have similar functions but because they are the only two modules that are not part of the input communications processing portion of the TIC. The Tactor Driver module receives the data and enable signal from the Command Processor and proceeds to calculate and output a new EI/RP cycle until it receives another command from the Command Processor. The Serial Feedback module receives the enable signal from the Auxiliary Command Processor and transmits data back to the microcontroller about the state of the TIC.

As indicated above, some of the principal modules have the ability to generate errors based on the format of the input data. If an error is generated, all modules except the Command Processor and Tactor Driver modules are reset to known initial states.

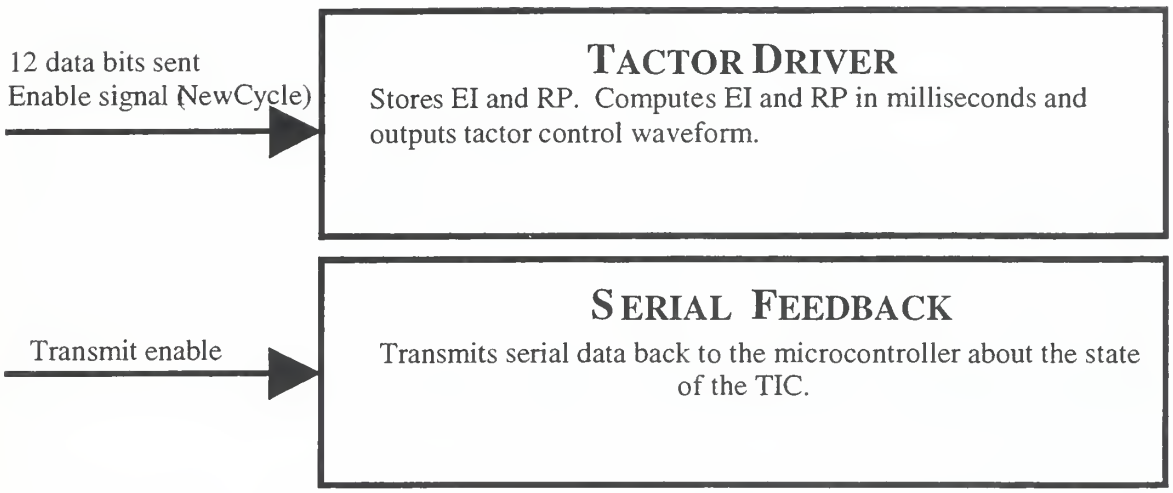



Figure 20. Tactor Driver and Serial Feedback modules

### 2. Detailed Principal TIC Module Descriptions

The TIC logic is designed for implementation in a single VLSI (Very Large Scale Integration) chip. For that reason, NAND and NOR gates were used as much as possible in place of AND and OR gates. In VLSI design, NAND and NOR gates require fewer transistors to construct than do their AND and OR counterparts [Ref. 8]. Additionally, some of the primary modules are complex state machines with multiple nested feedback loops. No claim is made in this document as to the optimality of these modules with respect to speed or minimal gate count. Karnaugh maps and other minimizing techniques were used wherever possible to optimize logic implementations, but as Wakerly himself admits [Ref. 9], it is impractical to use most logic design techniques on complex designs with multiple feedback loops. The schematics for the principal module designs are grouped in Appendix A for easy reference.

At the end of each detailed module description is a list of the secondary components used in the module. Secondary components are basic logic components discussed in most introductory texts on logic design. Examples include latches and synchronous counters. The secondary components have been optimized using standard design techniques to minimize the number of gates and make the transition to VLSI implementation easier. The secondary components are described in detail at the end of this chapter and the corresponding schematics can be found at the end of Appendix A.

Throughout the logic schematics, there are two symbols used for binary constants. The arrow pointing up labeled VCC is a constant binary one and the arrow pointing down labeled GND is a constant binary zero. The remainder of the symbols used in the TIC logic design are standard logic gate symbols with corresponding reference numbers [Ref. 9].

### *a. Serial Data Shift Register*

The Serial Data Shift Register, Figure 30, is a collection of ten edge-triggered D flip-flops (Dff) connected in series (Q outputs connected to D inputs) with a common active-low preset and common clock. The input to the first flip-flop in the series is the serial data arriving from the microcontroller. The clock signal for the bank of flip-flops is the clock for the USART communications link. Since the idle state in a UART is typically a binary one, the flip-flops in the shift register are preset to a binary one whenever the TIC is reset to its initial state. All ten Dff outputs and their complements are outputs of this module and are available for use by the other principal components. The only secondary component used in this module is the Dff.

### *b. Serial Data Protocol Check Module*

The inputs to this module, Figure 31, are the eight data bits and the parity bit from the shift register, the USART clock signal, and the reset signal called state0*. In this design, the asterisk after a signal name or a bubble on the body of a component means the signal is active low.

This component is essentially a Mealy state machine. The three states of the module are diagrammed in Figure 21 below. In the idle state 00, the first Dff of the Serial Data Shift Register is checked every clock cycle to see if a start bit (a binary zero) has been

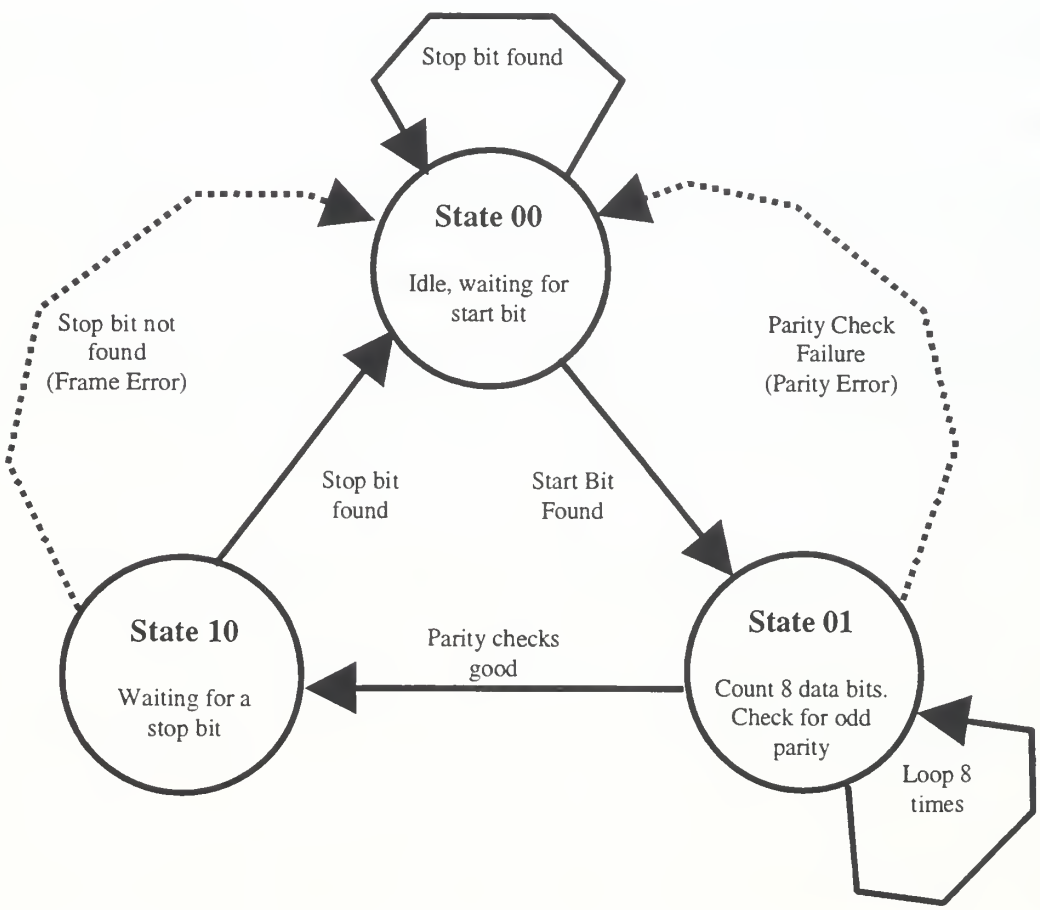



Figure 21. Serial Data Format Check module state diagram

sent. If no start bit is present, the module remains in the idle state. If the Dff contains a start bit, the state machine transitions to state 01. In state 01, a four-bit up counter is started which counts the passage of nine clock periods (eight data bits and a parity bit). The four-bit counter counts continuously regardless of the state of the module. The counter is asynchronously cleared to zero immediately after the start bit is detected.

When the counter arrives at nine, the parity of the first nine bits in the shift register is computed and, if the parity is odd, the state machine transitions to state 10. If the parity is even, a parity error flag is set and the state machine returns to the idle state. In state 10, the next bit expected to be shifted into the Serial Data Shift Register is a stop bit. If a stop bit is shifted into the shift register then a decode enable flag is sent to the Data Decode module. Otherwise, a framing error flag is set. In either case, the state machine returns to the idle state. The asynchronous parity error and frame error signals must be routed through a Dff before leaving the module because they are used as inputs to asynchronous clears of other modules and it facilitates meeting the setup and hold timing requirements of downstream flip-flops. The error signal and decode signal remain high for one clock cycle if either is activated.

The serial decoder module is the most critical module with regard to the speed of the communications link to the TIC. The length of the critical path within this module determines the maximum clock speed for the serial transmission line. In the following modules, the module states change at most once every eleven clock periods (the length of a USART frame), while state changes in this module occur in the space of one

clock period. It is therefore critical that this module be as efficient as possible in its design and implementation.

Secondary components used in this module are the State Machine, the four-bit synchronous up counter, the odd parity check module, and the Dff.

### *c. Data Decode Module*

The Data Decode module, Figure 32, is a series of latches and a three state Mealy machine. It is the boundary between the fast-clocked, bit-oriented Serial Protocol Check module and the remainder of the design, which is byte-oriented.

The state diagram for this module is shown in Figure 22. In the idle state 00, the module is waiting to receive an address that matches the pre-programmed address or the broadcast address (all 1's). Any other decoded byte is intended for another tactor and is ignored. When the module decodes a matching address, it moves to state 01, called the active state. If the module decodes another address while in state 01, it remains in state 01. This enables the microprocessor to address several tactors at once, putting them all into state 01 ("listening" for a command word) then send a single command to which all active tactors will respond.

As diagrammed earlier in this chapter, there are two command word formats. The first is an auxiliary command byte, denoted by the format $01c_5c_4c_3c_2c_1c_0$. If the module is in state 01 and receives an AC, it sets the AC flag indicating that the data in the latch is an AC then returns to state 00.

The second command word format is an Excitation Interval/Repetition Period command word pair. If the module is in state 01 and receives an EI (denoted by the format $00e_5e_4e_3e_2e_1e_0$), it moves to state 10, sets a flag indicating that the data in the latch is an EI, and waits for an RP. If it is in state 10 and receives an RP (denoted by the format $00p_5p_4p_3p_2p_1p_0$), the module sets a flag indicating that the data in the latch is an RP and moves to state 00. Since it makes no sense to address multiple tactors in the middle of an EI/RP command word pair, receiving an address in state 10 (denoted by the format $1a_6a_5a_4a_3a_2a_1a_0$) results in the setting of a Sequence Error flag and a return to state 00.

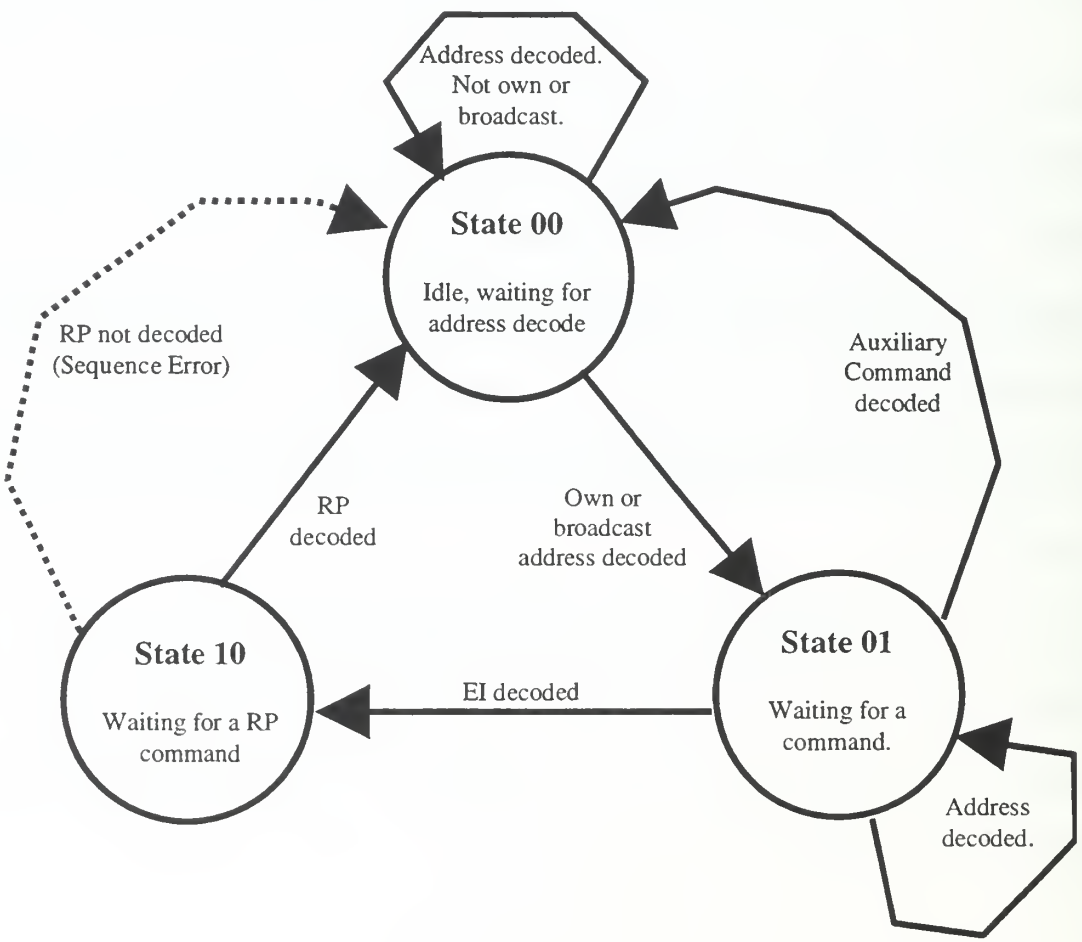



Figure 22. Data Decode module state diagram

When this module receives a decode enable pulse from the Serial Protocol Check module, a series of sequential events takes place which are independent of the current state of the module. These events must take place in order for the state transitions detailed above to occur. The three series-connected D-flip-flops on the left of the schematic act as a shift register and are clocked on alternating positive and negative clock edges to avoid race conditions in other parts of the module. The first Dff enables the 8-bit latch to store the data. The second Dff clocks the flip-flops on the right side of the schematic to store the outputs calculated with the new data in the latch. The third Dff changes the state of the module based on the value of the newly calculated outputs. The numbered list below details the operation of the module.

1. One half clock period after the decode enable signal is received from the Serial Data Protocol Check module (before the next serial data bit is shifted into the shift register), data bits 0-7 are latched into an 8-bit latch in the Data Decode module. The latch enables the Data Decode module to work with the data while the Serial Data Shift Register shifts in new data on the following positive clock edge.
2. Based on the value of the data in the 8-bit latch and the current state of the module, the module outputs and the next state are computed by combinational logic.
3. On the next falling clock edge, the module outputs are latched into five D-flip-flops on the right side of the schematic.
4. On the next rising clock edge, the State Machine is clocked to the next computed
   state.

The module checks for the correct address and for the broadcast address using XOR gates and inverters, since the design software used does not have an XNOR gate. Secondary components used in this module are the State Machine, 8-bit latch, and the Dff.

### *d. Command Processor Module*

The Command Processor module, Figure 33, acts as a buffer to separate the communications portions of the chip from the portion of the chip that drives the tactor. Errors in communications such as parity, frame, and sequence errors have no effect on this module or the Tactor Driver module. This module is also the boundary between those modules that use the USART clock (high frequency) and those modules that use the tactor driver clock (low frequency). Although the EI, RP, and AC data is contained in the first six bits of the command words, all eight bits are latched into this module to make the chip more generic and for ease of future command word modification or expansion.

The Command Processor module contains three separate 8-bit latches. When an EI or AC enable signal is received, the eight-bit EI or AC data word is latched into Latch A. If an RP enable signal is received, the data in Latch A (an EI data word) is latched into Latch B and the RP data word is latched into Latch C. Combinational logic determines if the new EI/RP pair is different than that which is currently driving the tactor output. If it is a different command pair, a New Cycle flag is generated and sent to the Tactor Driver module. If not, the current tactor cycle is not disturbed. Latches B and C drive the counters in the Tactor Driver module and directly affect the output waveform. Latch A isolates latches B and C from sequence errors and enables auxiliary commands to be executed without disturbing ongoing tactor waveform generation.

Two arrays of XOR gates determine if the new EI/RP data is different than what is currently in Latches B and C, and sets the NewCycle flip-flop accordingly. Secondary components used in this module are the eight-bit latch and the Dff.

### *e. Auxiliary Command Processor Module*

This module, Figure 34, is nothing more than combinational logic that decodes 6-bit Auxiliary Commands. For all practical purposes, it is merely an extension of the Command Processor module. The combinational logic is enabled by the AC enable signal from the Word Decoder module. The data to be decoded is taken from Latch A in the Command Processor module. The Auxiliary Command Processor module is the only module that has to be modified to add additional TIC functionality such as variable voltage adjustment, frequency adjustment, test mode commands, etc. Currently, this module decodes only two commands, Reset (100000) and Transmit Feedback (100001). The output of this module is a Reset signal, and a Transmit Feedback signal. The bits and their complements not used in the decoding of the reset and transmit signals are terminated in inverters for future use. The only secondary component used in this module is the Dff.

### *f. Serial Data Feedback Module*

This module, Figure 35, is very simple but can be extremely important for hardware and software debugging once the design is implemented in hardware using either TTL (Transistor-Transistor Logic) chips, a CPLD (Complex Programmable Logic Device), or a VLSI chip. Using the serial feedback module, the designer can have immediate access to all internal signals that are connected to the feedback module for monitoring correct chip

operation, correct tactor operation, and error catching. The usefulness of this module is limited only by the creativity of the designer and end user. The inputs to this module are a Transmit Enable signal from the Auxiliary Command Processor module, the USART clock, a reset signal (state0*), and 32 individual signal lines, which can be connected to any signals in the TIC. These 32 signal lines are inputs to three 16x1 multiplexers, the outputs of which are in turn connected to a 4x1 multiplexer. Using a 6-bit counter, each input of the three multiplexers is sequentially accessed (one bit per clock cycle) and transmitted serially using a flip-flop as a transmitter. The signal lines are connected to the multiplexers such that they are correctly framed by a start bit, parity bit, and a stop bit. The parity bit is generated using the same secondary module used to check for odd parity in the input serial data stream.

The transmit signal from the Auxiliary Command Processor module clocks a binary one into a flip-flop in the feedback module which starts a 6-bit up counter. The outputs of the counter are connected to the select inputs of the multiplexers. The outputs of the three 16x1 multiplexers are connected to what is, essentially, a 3x1 multiplexer. As the counter counts up, each multiplexer input is selected in turn and placed on the transmit line by the flip-flop on the right side of the schematic. The start and stop bits are embedded in the multiplexer inputs at the appropriate places. The parity bit generator is the same 9-bit odd parity checker used in the Serial Decoder module, with the ninth bit tied high. When the counter reaches the end of the third 16x1 multiplexer, the counter is cleared and the enabling flip-flop is cleared. When the counter is at zero, the first bit of the first multiplexer is continuously being transmitted. This bit is always a binary one and represents the idle

state in UART communications. Absence of voltage on the feedback line is an indication of a problem with the TIC.

In this design, only the two 6-bit EI and RP words are wired to the feedback transmitter and the rest of the signal lines are reserved for future use. The capability to monitor the EI/RP data is useful to ensure the data sent to the tactor actually makes it through the TIC error-free. Examples of other useful signals to connect to the feedback transmitter are module states, error flags, last command executed, and tactor status (on/off, melted, missing, etc.). A good design idea is to activate the feedback transmitter in the event of any error in order to transmit the error type and module status. This would require some priority-encoding scheme to ensure no two tactors transmitted on the feedback line at the same time. Secondary components used in this module are a 6-bit synchronous up counter, odd parity check module, 16x1 multiplexer, and the Dff.

#### *g. Tactor Driver Module*

This module, Figure 36, is the heart of the TIC. All other modules exist to support the Tactor Driver module. The basic concept behind the module is fairly simple, but the implementation is relatively complex to ensure exact duty cycle and repetition period timing as well as correct transitions between different timing cycles. The behavioral steps for this module are as follows.

1. When the NewCycle flag is received from the Command Processor module, two 6-bit counters are loaded with the bits stored in latches B and C (EI and RP) from the Command Processor module.
2. If the RP is zero, the tactor is turned off. If the RP is not zero and the EI is zero, the tactor is turned on.

3. If neither RP nor EI are zero, the tactor is turned on and the EI and RP counters begin to count down.
4. When the EI counter reaches zero, the tactor is turned off. Some period of time later when the RP counter reaches zero, the tactor is turned on and the two six-bit counters are again loaded with the EI and RP values from Latches B and C.
5. The counters begin to count down again (repeating the cycle).

It should be obvious the RP value must always be greater than the EI value. This results in several invalid EI/RP binary combinations. Although having so many invalid combinations may seem like a waste of bits, this scheme is simple to implement in hardware. The tradeoff in reduced gate count is worth the slight increase in transmitted bits. Also, there is no error checking in the hardware to catch invalid combinations. All error checking for valid EI/RP combinations must be performed in software.

To meet the desired EI/RP timing specifications detailed earlier in this chapter, the EI and RP counters must be clocked at specific frequencies. Since the skin is most sensitive to a frequency of approximately 250 Hz [Ref. 3], the tactor is driven with a 250 Hz carrier wave. The Nyquist sampling theorem requires the frequency for the modulating wave can never be greater than 125 Hz for accurate waveform generation. This is not a problem since the average person cannot discriminate between frequencies 100 Hz and greater [Ref. 3]. Since a 250 Hz wave is used to drive the tactor, 250 Hz is also the logical choice for a clock to drive the counters. Thus, the maximum timing resolution for the counters is four milliseconds. For the EI counter, the 250 Hz clock is divided by four using a four-bit counter, resulting in an EI counter clock of 62.5 Hz with a 16 millisecond period. The six-bit EI value has a decimal range of 1-63 multiplied by 16 milliseconds which yields a range of Excitation Interval time of 16-1008 milliseconds with a resolution

of 16 milliseconds. Likewise for the RP counter, the 250 Hz clock is divided by 16 using the four-bit counter. The result is a RP counter clock of 15.620 Hz with a 64 millisecond period. The six-bit RP value also has a decimal range of 1-63 multiplied by 64 milliseconds which yields a range of Repetition Period time of 64-4032 milliseconds with a resolution of 64 milliseconds. These numbers meet the tactor specifications given at the beginning of this chapter.

The four-bit counter and two six-bit counters are easy to recognize in the center of the schematic. The NOR gates above and below the six-bit counters check for the all-zero RP and EI data words. If both are presented at the same time, the all-zero RP that turns the tactor off overrides the all-zero EI that turns the tactor on. The eight-input NAND gates that signal the counter has reached zero are actually eight-input NAND gates with two unused inputs. Six-input NAND gates do not exist in the design software. The five flip-flops on the right side of the schematic introduce the correct amount of delay in the start-tactor and stop-tactor signals to ensure proper timing of the modulating envelope and the reloading of the counters. Much of the timing adjustment of the module was accomplished using the trial and error method. Secondary components used in this module are the four-bit loadable synchronous down counter, six-bit loadable synchronous down counter, and the Dff.

### 3. Secondary Modules

Schematics of these modules are located in Appendix A after the primary module schematics, Figure 37 - Figure 46. The secondary modules are the State Machine, odd parity check/generator, eight-bit latch, four-bit synchronous up counter, six-bit synchronous

up counter, four-bit loadable synchronous down counter, six-bit loadable synchronous down counter, 16x1 bit multiplexer, edge-triggered D-flip-flop, and edge-triggered T-flip-flop. These basic designs have been optimized to reduce the number of logic gates and make the transition to VLSI implementation easier.

## D. HARDWARE IMPLEMENTATIONS

The detailed design presented in this chapter thus far is merely the logical design of a TIC. As mentioned previously, it is one of many such designs and implements advanced features that may be omitted from initial hardware implementations. The choice of hardware to implement this design is an important one. Three hardware implementations are examined in this section: TTL (Transistor-Transistor Logic), programmable logic, and a single-chip VLSI implementation.

### 1. TTL Implementation

The TTL implementation is the least expensive of the three implementations. Each 7400-series TTL chip costs only pennies and contains anywhere from one to eight logic gates. Furthermore, TTL chips usually can be found in any digital logic design laboratory or at convenient neighborhood stores such as Radio Shack. Unfortunately, the complexity of the TIC design as presented in this chapter requires 302 TTL chips. The area required by the chips and wiring is approximately 96 square inches, making the design implementation prohibitively large. The TTL implementation would be adequate to test the logic design for logic hazards before choosing a more expensive implementation, but since most design

software performs the same checking functions the TTL implementation is rather unnecessary.

### 2. Programmable Logic Implementation

Programmable logic is loosely defined as a device with configurable logic and flip-flops linked together with programmable interconnect. Memory cells define the function that the logic performs and how the various functions are interconnected. Though various devices use different architectures, all are based on this fundamental idea. There are a few major programmable logic architectures available today. Each architecture typically has vendor-specific sub-variants within each type. The major types include:

1. Simple Programmable Logic Devices (SPLDs)
2. Complex Programmable Logic Devices (CPLDs)
3. Field Programmable Gate Arrays (FPGAs)

SPLDs are the smallest and consequently the least-expensive form of programmable logic. An SPLD is typically comprised of four to 22 macrocells and can typically replace a few 7400-series TTL devices. A macrocell usually contains a sum-of-products combinatorial logic function and an optional flip-flop. The combinatorial logic function typically supports four to sixteen product terms with wide fan-in. In other words, a macrocell may have many inputs, but the complexity of the logic function is limited. Contrast this to most FPGA logic blocks where the complexity is virtually unlimited, but the logic function has just four inputs. Each of the macrocells is typically fully connected to the others in the device. Most SPLDs use either fuses or non-volatile memory cells such as

EPROM (Erasable Programmable Read Only Memory), EEPROM (Electronically EPROM), or Flash to define the functionality.

The CPLDs are also known as EPLD (Erasable Programmable Logic Device), and EEPLD (Electrically-Erasable Programmable Logic Device). CPLDs are similar to SPLDs except they have a significantly higher capacity. A typical CPLD is the equivalent of two to 32 SPLDs. A CPLD typically contains from 18 to 256 macrocells and can typically replace tens of 7400-series TTL devices. A group of eight to 16 macrocells is typically grouped together into a larger function block. The macrocells within a function block are usually fully connected. If a device contains multiple function blocks, then the function blocks are further interconnected. Not all CPLDs are fully connected between function blocks; this is vendor and family specific. Less that 100% connection between function blocks means there is a chance that the device will not route or may have problems keeping the same pinout between design revisions. CPLDs are generally best for control-oriented designs, due in part to their fast pin-to-pin performance. The wide fan-in of their macrocells makes them well suited for complex, high performance state machines. CPLDs are manufactured using one of three process technologies: EPROM, EEPROM, or Flash. EPROM-based CPLDs are usually one-time programmable (OTP) unless they are in an ultraviolet (UV) erasable windowed package. A device programmer or the manufacturer or distributor programs an EPROM-based CPLD. Generally, CPLDs use non-volatile memory cells such as EPROM, EEPROM, or Flash to define the functionality. Many of the new CPLD families use an EEPROM or Flash and have been designed so they can be programmed in-circuit (also called ISP for in-system programmable).

FPGAs are also known as LCA (Logic Cell Array), pASIC (programmable ASIC), FLEX (Altera), ACT (Actel), and ORCA (Lucent). FPGAs are distinct from SPLDs and CPLDs and typically offer the highest logic capacity. An FPGA consists of an array of logic blocks, surrounded by programmable I/O blocks, and connected with programmable interconnect. A typical FPGA contains from 64 to thousands of logic blocks and an even greater number of flip-flops. Most FPGAs do not provide 100% interconnect between logic blocks (to do so would be prohibitively expensive). Instead, sophisticated software places and routes the logic on the device much like a PCB autorouter would place and route components.

A generic description of an FPGA is a programmable device with an internal array of logic blocks, surrounded by a ring of programmable input/output blocks, connected together via programmable interconnect. There are a wide variety of sub-architectures within this group. The secret to density and performance in these devices lies in the logic contained in their logic blocks and on the performance and efficiency of their routing architecture.

Clearly, either a CPLD or an FPGA would be a good choice for a hardware implementation for the TIC design. Programmable logic chips are more expensive than TTL chips, but the size of the programmable chips is much smaller than the TTL chips. Most high-density programmable logic devices on the market today are about 1 square inch in size. This is still much too large for the incorporation into a 40-tactor TSAS vest, but programmable logic is an excellent hardware implementation to test and debug the TIC logic design. A reprogrammable FPGA would enable the designer to make changes to the

logic and to see the results almost immediately. However, programmable logic comes with two major drawbacks: size and programmability. Although much smaller than the TTL chips, 1 square inch per tactor times 40 tactors is still too large to be comfortably worn under a flight suit. Additionally, programmable logic requires expensive vendor-specific equipment in order to program the logic devices.

### 3. Very Large Scale Integration Implementation

VLSI circuits have permeated the electronics industry in the form of consumer and commercial chips such as microprocessors, memory, digital signal processors, and embedded controllers. These mass-market chips have found their way into an astounding range of consumer products from computers down to simple appliances such as toasters and vacuum cleaners.

A second wave of VLSI chips is now becoming prevalent in many commercial ventures. These application-specific integrated circuits (ASICs) are, by their nature, designed for specialized applications and produced in much lower quantities than the previously described chips. In order to make such designs economically feasible given their limited production quantities, a VLSI design methodology has evolved to greatly reduce the design time while preserving high performance levels. Combined with the availability of low-cost tools and inexpensive fabrication prototyping, this methodology facilitates the design of analog, digital, and mixed-signal ASICs (Application Specific Integrated Circuits) by anyone from large corporations to individual VLSI designers. [Ref. 8]

The size of the VLSI chip makes it the ideal (and only) choice for the hardware implementation of the tactor array TIC logic design. Unfortunately, with the smaller size come lengthened design cycles and increased costs over the previous two hardware implementations. LCDR Jeffrey Link is currently researching a TIC VLSI hardware implementation [Ref. 10].

# IV. THE MICROCONTROLLER SYSTEM DESIGN

The Tactor Interface Microcontroller System (TIMS) has two main components. The first is the Intelligent Tactor that was discussed in the previous chapter. The second is the microcontroller system that is discussed in this chapter. The purpose of the microcontroller system is to collect sensor information from external sources, analyze the information, and communicate with the TICs to display the information on the tactor array.

The central processing unit of the microcontroller system is the Motorola MPC860. The MPC860 supplies the control and processing power to run all the other microcontroller system components. The system includes a MIL-STD-1553 bus interface to communicate with modern aircraft computer systems and standard serial interfaces to communicate with the TICs and other serial devices. The system also includes memory, reset circuitry, power and clock circuitry, and a debug port.

The remainder of this chapter provides detailed information about the microcontroller system including specifics on the MPC860 that are relevant only to a software or hardware designer. The microcontroller system schematics corresponding to the sections in this chapter are grouped in Appendix B for convenient reference. Example schematics from the MPC860 Application Development System (ADS) [Ref. 11] and the MPC860 Reference Design Board (SAMBA) Release Guide [Ref. 12] were used extensively as a guide for the design shown in Appendix B.

## A. MOTOROLA MPC860 MICROCONTROLLER

The Motorola MPC860 PowerPC Quad Integrated Communications Controller (PowerQUICC) [Ref. 13] is a versatile single-chip integrated microprocessor and peripheral combination that can be used in a variety of controller applications, excelling particularly in communications and networking products. Unless otherwise specified, the PowerQUICC chip is referred to as the MPC860 in this document. The MPC860 is a PowerPC architecture-based derivative of Motorola's MC68360 Quad Integrated Communications Controller (QUICC), referred to here as the QUICC. The PowerQUICC can be described as a next-generation MC68360 QUICC for network and data communication applications, providing higher performance in all areas of device operation including flexibility, extensions in capability, and integration. The MPC860 PowerQUICC, like the MC68360 QUICC, integrates two processing blocks. One block is the Embedded PowerPC Core and the second block is a Communication Processor Module (CPM) that closely resembles the MC68360 CPM. This dual-processor architecture provides lower power consumption than traditional architectures because the CPM off-loads peripheral tasks from the Embedded PowerPC Core.

While not all of the MPC860 functions detailed in the following sections are utilized in the microcontroller system design detailed in this chapter, it is nonetheless useful to include them in this section as a reference for future hardware and software designers. Furthermore, the many used and unused functions of the MPC860 demonstrate the ability of the chip to easily expand to fit the growing needs of tactile technology.

## 1. Overview

The CPU on the MPC860 is a 32-bit PowerPC implementation that incorporates memory management units (MMUs) and instruction and data caches. The communications processor module from the MC68360 QUICC has been enhanced by the addition of the inter-integrated controller ($I^2C$) channel. Digital signal processing (DSP) functionality has been added to the CPM. The memory controller has been enhanced, enabling the MPC860 to support any type of memory, including high-performance memories and new types of DRAMs. A PCMCIA (Personal Computer Memory Card International Association) socket controller supports up to two sockets. A real-time clock has also been integrated. A block diagram of the MPC860 is shown in Figure 23. With the exception of the performance subsection, all figures and MPC860 functional descriptions in this overview were reproduced or paraphrased from Reference 13.

The MPC860 PowerQUICC integrates the Embedded PowerPC Core with high performance, low power peripherals to extend the Motorola Data Communications family of embedded processors even farther into high end communications and networking products. The MPC860 PowerQUICC is comprised of three modules which all use the 32-bit internal bus: the Embedded PowerPC Core, the System Interface Unit (SIU), and the Communication Processor Module (CPM).

A fundamental design goal of the MPC860 PowerQUICC is ease of interface to other system components. Figure 24 shows a system configuration that offers one EPROM, one flash EPROM, and supports two DRAM (Dynamic Random Access Memory) SIMMs (Single In-line Memory Modules). Depending on the capacitance on the system bus,

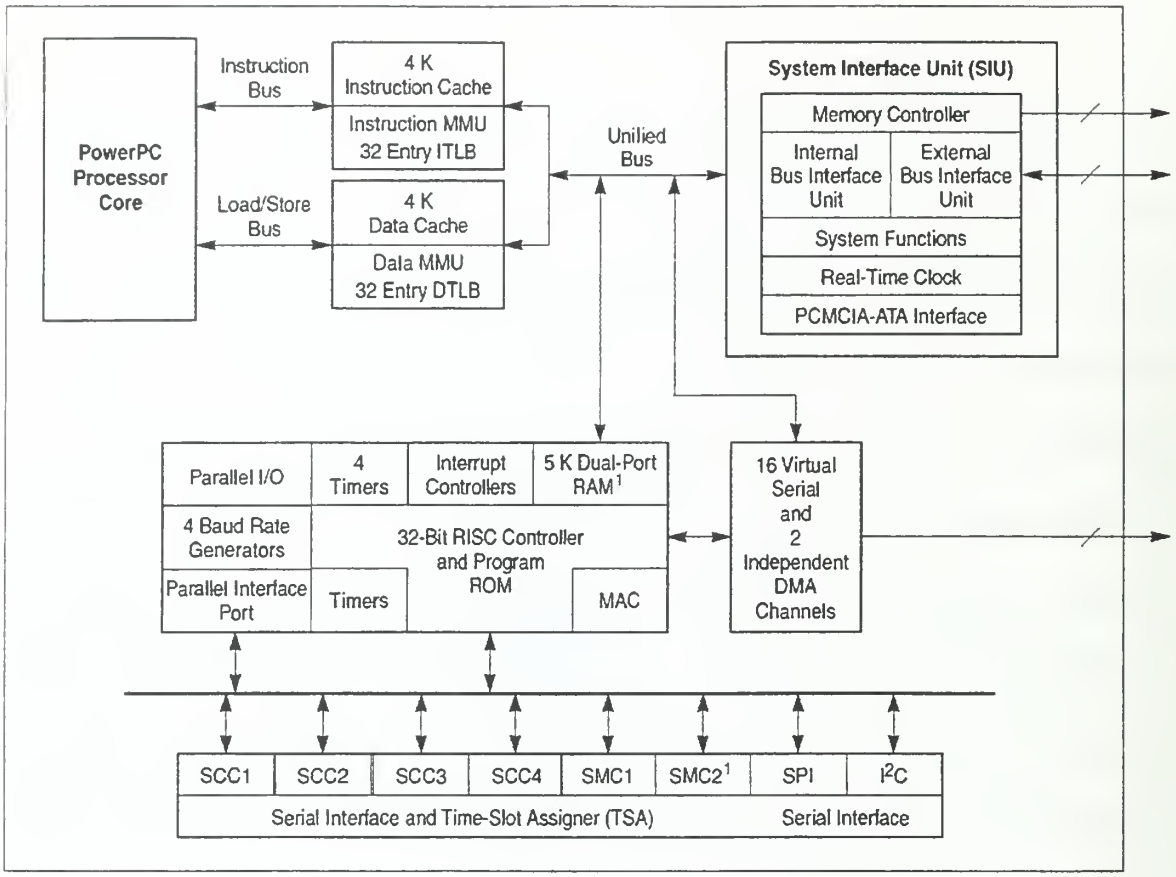



Figure 23. MPC860 block diagram [Ref. 13]

external buffers may be required. From a logic standpoint, however, a glueless system is maintained.

### *a. Communications Processor Module*

The MPC860 PowerQUICC implements a dual processor architecture. This dual processor architecture provides both a high performance general-purpose processor for application programming use as well as a special purpose communication processor uniquely designed for communications needs.

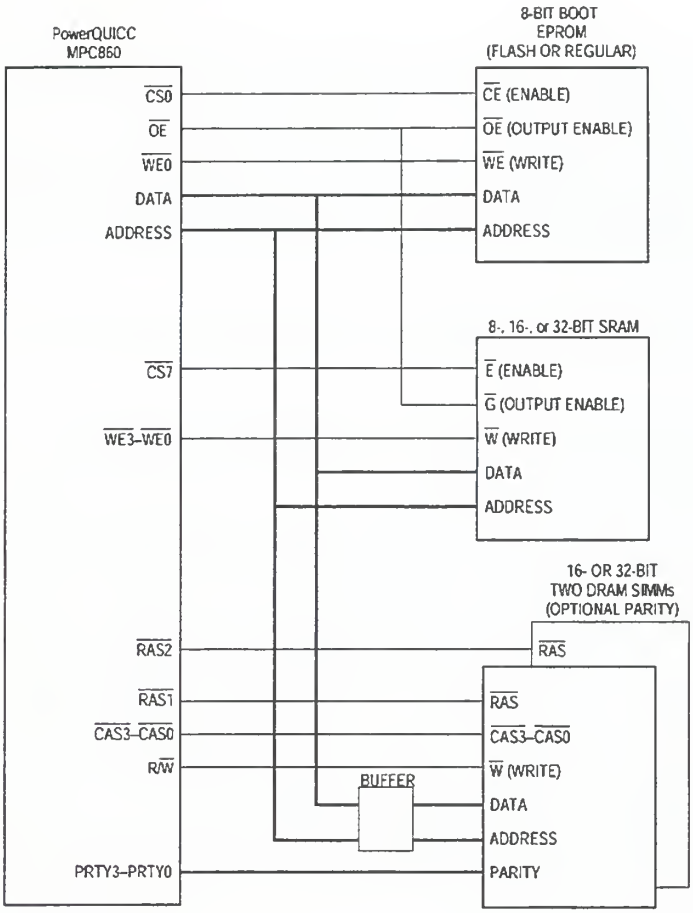



Figure 24. MPC860 memory connection example [Ref. 13]

The CPM provides a flexible and integrated approach to communications-intensive environments. To reduce system frequency and save power, the CPM has its own independent RISC (Reduced Instruction Set Computer) communications processor (CP) that is optimized for serial communications. The CP services several integrated communications channels, performing low-level protocol processing and controlling DMA (Direct Memory Access). The CPM supports multiple communications channels and protocols, and it has flexible firmware programmability. The CPM frees the core of many computational tasks in the following ways [Ref. 13]:

1. By reducing the interrupt rate. The core is interrupted only upon frame reception or transmission, instead of on a per-character basis.
2. By implementing some of the OSI (Open Systems Interconnection) layer-2 processing, which provides more core bandwidth for higher layer processing.
3. By supporting multibuffer memory data structures that are convenient for software handling.

The features of the CPM may be divided into three sub-groups [Ref. 13]:

1. Communications Processor (CP)
2. Sixteen Independent DMA (or serial DMA) Controllers
3. Four General-Purpose Timers

Transacting with the communications peripherals on a separate bus from the PowerPC core, the CPM's 32-bit communications processor handles the low-level communications tasks, freeing the core for higher-level tasks. The CP provides the communication features of the MPC860 PowerQUICC. Included are a RISC processor, four Serial Communication Controllers (SCC), four Serial Management Controllers (SMC), one Serial Peripheral Interface (SPI), one $I^2C$ Interface, 5 kB of dual-port RAM (Random Access Memory), an interrupt controller, a time slot assigner, three parallel ports, a parallel interface port, four independent baud rate generators, and sixteen serial DMA channels to support the SCCs, SMCs, SPI, and $I^2C$.

The CP implements the chosen protocols using the serial controllers and parallel interface port and manages the data transfer through the serial DMA (SDMA) channels between the I/O channels and memory. It also manages IDMA (independent DMA) channels. The SDMAs provide two channels of general-purpose DMA capability for each communications channel. They offer high-speed transfers, 32-bit data movement, buffer chaining, and independent request and acknowledge logic.

The four general-purpose timers on the CPM support the internal cascading of two timers to form a 32-bit timer. The addition of a Multiply-And-Accumulate (MAC) function on the CPM further enhances the MPC860 PowerQUICC, enabling various modem and DSP applications. The following is a summary of the CPM's main features [Ref. 13].

1. Communications processor (CP)
    a. Dual-port RAM
    b. Internal ROM (Read Only Memory)
    c. DSP functions with 16-bit multiply/accumulate hardware (MAC)
    d. DMA control for all communications channels
    e. Two independent DMA channels for memory-to-memory transfers or interfacing
    f. External peripherals
    g. RISC timer tables
2. Four full-duplex serial communications controllers (SCCs) that support the following:
    a. UART protocol (asynchronous or synchronous)
    b. HDLC protocol
    c. Appletalk protocol
    d. Asynchronous HDLC protocol
    e. BISYNC (Binary Synchronous Communications) protocol
    f. Transparent protocol
    g. Infrared Data Association protocol (IrDA)
    h. IEEE 802.3 / Ethernet protocol
3. Two full-duplex serial management controllers (SMCs)
    a. UART protocol
    b. Transparent protocol
    c. GCI protocol for monitor and C/I channels (for ISDN)
4. Serial peripheral interface (SPI) support for master or slave modes
5. Inter-integrated circuit (I 2 C) bus controller
6. Parallel interface port (supporting Centronics)
7. A serial interface (SI) with a time-slot assigner (TSA) that supports multiplexing of data from SCCs and SMCs onto two time-division multiplexed (TDM) interfaces
8. Four independent baud rate generators (BRGs)
9. Four general-purpose 16-bit timers or two 32-bit timers
10. CPM interrupt controller (CPIC)
11. General-purpose I/O ports

### *b. System Interface Unit*

The system interface unit controls system startup, initialization and operation, protection, as well as the external system bus. The system configuration and protection function controls the overall system and provides various monitors and timers, including the bus monitor, software watchdog timer, periodic interrupt timer (PIT), PowerPC decrementer, timebase, and real-time clock. The clock synthesizer generates the clock signals for other modules and external devices with which the SIU interfaces. The SIU supports various low-power modes that supply different ranges of power consumption, functionality, and wake-up time. The clock scheme supports low-power modes for applications that use baud rate generators and/or serial ports in standby mode. The main system clock can be changed dynamically; the baud rate generators and serial ports work with a fixed frequency. The external bus interface handles the transfer of information between internal buses and the memory or peripherals in the external address space. The MPC860 is designed to allow external bus devices to request and obtain system bus mastership. The memory controller module provides a glueless interface to many types of memory devices and peripherals; it supports a maximum of eight memory banks, each with its own device and timing attributes. Memory control services are provided to both internal and external masters. The MPC860 supports circuit board test strategies through user-accessible test logic that is fully compliant with the IEEE (Institute of Electrical and Electronics Engineers) 1149.1 standard. [Ref. 13]

Although the Embedded PowerPC Core is always a 32-bit device internally, it may be configured to operate with an 8-, 16- or 32-bit data bus. Regardless of the choice of the system bus size, dynamic bus sizing is supported. Bus sizing allows 8-, 16-, and 32-bit peripherals and memory to exist in the 32-bit system bus mode. The memory controller will support up to eight memory banks with glueless interfaces to DRAM, SRAM (Static RAM), EPROM, Flash EPROM, EDO (Extended Data Out RAM) and other peripherals with two-clock access to external SRAM and bursting support. It provides variable block sizes from 32 kilobytes to 256 megabytes. The memory controller will provide 0 to 15 wait states for each bank of memory and can use address type matching to qualify each memory bank access. It provides four byte enable signals for varying width devices, one output enable signal and one boot chip select available at reset.

The PCMCIA host adapter module provides all control logic for a PCMCIA interface. This interface complies fully with the PCMCIA standard, Release 2.1+ (PC Card - 16). It can support two PCMCIA sockets with a maximum of eight memory or I/O windows. The following is a list of the SIU's main features [Ref. 13]:

1. System configuration and protection
2. System interrupt configuration
3. System reset monitoring and generation
4. Clock synthesizer
5. Power management
6. Real-time clock
7. PowerPC decrementer
8. Time base
9. Periodic interrupt timer (PIT)
10. External bus interface control
11. Eight memory banks supported by the memory controller
12. Debug support

13. PCMCIA host adapter module supports two slots with eight memory or I/O windows
14. IEEE 1149.1 test access port

### *c. Clocking and Power Requirements*

The MPC860 clock system provides many different clocking options for all on-chip and external devices. For its clock sources, the MPC860 contains phase-locked loop and crystal oscillator support circuitry. The phase-locked loop circuitry can be used to provide a high-frequency system clock from a low-frequency external source. Also, to enable flexible power control, the MPC860 provides frequency dividers and a variety of low-power mode options.

The MPC860 allows a system to optimize power utilization by providing performance on-demand. This is implemented through a variety of programmable power-saving modes with automatic wake-up features. The main features of the MPC860 clocks and power control system are as follows [Ref. 13]:

1. Contains system PLL (Phase Lock Loop)
2. Supports crystal oscillator circuits
3. Clock dividers are provided for low-power modes and internal clocks
4. Contains five major power-saving modes
    a. Normal (high and low)
    b. Doze (high and low)
    c. Sleep
    d. Deep sleep
    e. Power down

The various modules of the MPC860 are connected to four distinct power rails. These power rails have different requirements, as explained in the following paragraphs. The organization of the power rails is shown in Figure 25.

The I/O buffers, logic, and clock control are fed by a 3.3V power supply called VDDH. The internal logic can be fed by the same 3.3V source that powers VDDH, called VDDL. VDDL is identified as a separate power supply only to facilitate power measurements. VDDH must in all cases be greater than or equal to VDDL. To improve stability, the power supply pins for the SPLL (VDDSYN, VSSSYN, and VSSSYN1) are uniquely identified in order to allow special filtration to be provided for them. A well-regulated voltage should be applied to VDDSYN via a low impedance path to the

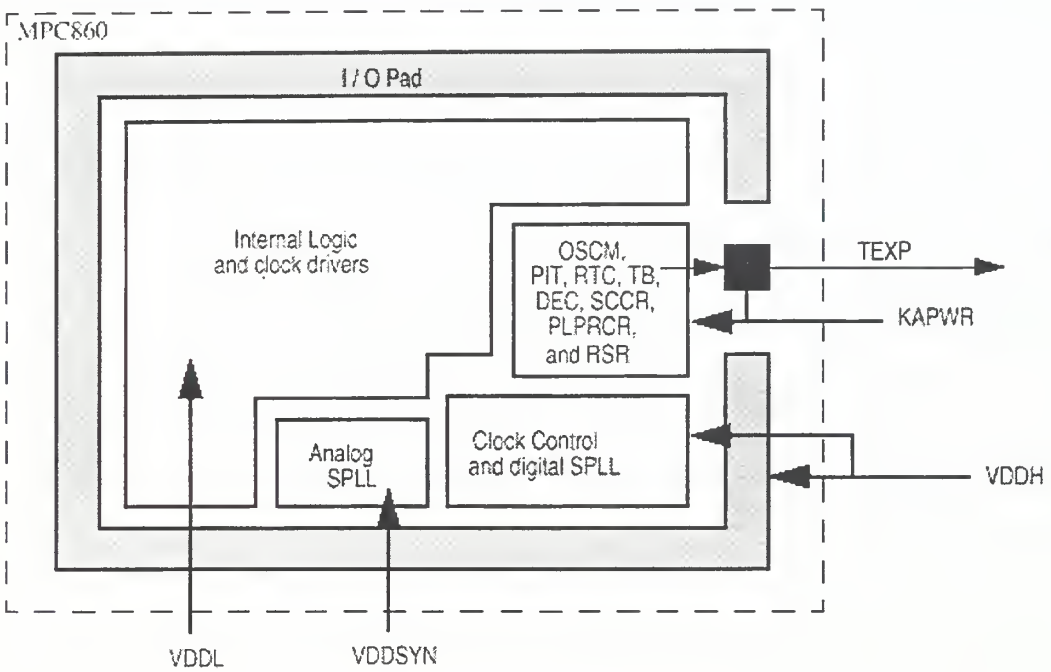



Figure 25. MPC860 power rails [Ref. 13]

VDDH/VDDL power rail. The allowable noise on the VDDSYN power plane is 20 mV peak up to a bandwidth of 100 MHz. This typically requires isolation of the VDDSYN power plane from the VDDH/VDDL power plane. An example implementation of this is a split power plane, with the VDDSYN plane implemented as an island in the VDDH/VDDL

power plane, connected to the VDDH/VDDL power plane with an inductor and to the ground plane with bypass capacitors. An inductor value of 8.2 mH and bypass capacitor values of 0.1 mF and 10 mF provide a two-pole filter with a cutoff frequency of 500 Hz. VSSSYN and VSSSYN1 must have a low impedance path to the ground plane. If sufficient isolation is provided for VDDSYN (as described above), no additional isolation for VSSSYN and VSSSYN1 is required. [Ref. 13]

The timebase, decrementer, periodic interrupt timer, and real-time clock are all connected to the keep-alive power (KAPWR) rail. This power rail architecture allows the system to remove the power at the VDDH/VDDL/VDDSYN pins during power-down mode. When VDDH is active, the internal modules connected to KAPWR are instead powered by VDDH. KAPWR is only used for this function when power at VDDH is shut off. This operation conserves the power of the KAPWR supply.

To optimize power consumption, the MPC860 provides low-power modes that can be used to dynamically activate and deactivate certain internal modules, such that only the needed modules are operating at any given time. In addition to normal high mode (i.e. fully activated), the MPC860 supports normal low, doze high, doze low, sleep, deep-sleep, and power-down modes.

In addition to these power-saving modes, it should be noted that the architecture of the CPM, described earlier in this chapter, inherently supports optimum power consumption. When the CPM is idle, it uses its own power-saving mechanism to shut down automatically.

### *d. Performance*

The performance of a single function chip such as a memory chip is relatively easy to quantify using such measures as access time. In contrast, the performance of a single chip microprocessor system or a desktop computer system is much harder to describe and is not, as some advertisements seem to indicate, merely a function of clock speed. The only true performance measure of any given computer system is the execution time for the task for which it is used in practice. Since it is impractical to test each computer system independently and use the results to compare the performance of different processors, a series of benchmark tests have been designed to simulate common tasks performed by typical computer systems.

A benchmark is test that measures the performance of a system or subsystem on a well-defined task or set of tasks. Benchmarks are commonly used to predict the performance of an unknown system on a known, or at least well-defined, task or workload. Benchmarks can also be used as monitoring and diagnostic tools. By running a benchmark and comparing the results against a known configuration, one can potentially pinpoint the cause of poor performance. Similarly, a developer can run a benchmark after making a change that might impact performance to determine the extent of the impact. Benchmarks are frequently used to ensure the minimum level of performance in a procurement specification. Rarely is performance the most important factor in a purchase, though. One must never forget that it's more important to be able to do the job correctly than it is to get the wrong answer in half the time.

The Dhrystone benchmark program [Ref. 14] has become a popular benchmark for CPU/compiler performance measurement, in particular in the area of minicomputers, workstations, PC's and microprocessors. It apparently satisfies a need for an easy-to-use integer benchmark; it gives a first performance indication, which is more meaningful than MIPS (Million Instructions Per Second) numbers which, in their literal meaning (million instructions per second), cannot be used across different instruction sets (e.g., RISC vs. CISC).

Dhrystone version two is a short synthetic benchmark program intended to be representative for system (integer) programming. Version one is no longer recommended since state-of-the-art compilers can eliminate too much 'dead code' from the benchmark (However, quoted MIPS numbers are often based on version one). Due to its small size (1-1.5 KB code), the memory system outside the cache is not tested and string operations are somewhat over-represented.

The performance Database Server at the Naval Ocean Systems Center in San Diego, California maintains a database of over 325 systems that have all been tested against most of the benchmarks in use today and ranked according to performance. Using the Dhrystone V2.1 benchmark, the 50MHz MPC860 achieves 66 MIPS, and the 66 MHz MPC860 achieves 87 MIPS. When compared against the Performance Database Server list, 66 MIPS is roughly equivalent to a Pentium P5-75 MHz system running Windows 95, or a SPARCstation 10/51 with a SuperSPARC CPU running SunOS 4.1.3. The performance of the 87 MIPS chip is roughly equivalent to a Pentium P5-100 MHz system running Windows 95, or a SPARCstation 5 with a uSPARCII CPU running Solaris 2.5 [Ref 15].

Although no benchmark should be blindly trusted, the numbers presented in this paragraph indicate the relative processing power of the given systems for the types of applications in which the TIMS will be used.

### 2. MPC860 Microcontroller Design Implementation

As noted earlier in this chapter, the schematics for the microcontroller design are grouped in Appendix B for convenient reference. The MPC860 connections can be found in Figure 47. The symbol for the MPC860 is in the center of the diagram and has several noteworthy features common to all symbols in the design. The alphanumeric symbols surrounding the symbol correspond to the physical pin numbers on the MPC860 chip. The part identifier is in the center of the MPC860 symbol and is always preceded by a "U." In this case, the part identifier is U7, which is a reference to the MPC860 part in the PCB layout in Appendix C and in the parts list in Appendix D. The signals that interface to other schematics in Appendix B are indicated with a "\I" after the signal name. Signals which have <0..31> after the signal names indicate that the single wire represents a bus, in this example a 32-signal bus. Each curved line that taps a signal off the bus is labeled with the signal number that is being tapped.

## B. MIL-STD-1553 INTERFACE

The MIL-STD-1553 Aircraft Internal Time Division Command/Response Multiplex Data Bus, [Ref. 16], is a military standard which has become one of the basic tools being used today by the DoD for integration of aircraft and weapon systems. The standard describes the method of communication and the electrical interface requirements for

subsystems connected to the data bus. The 1 Mbps serial communication bus is used to achieve aircraft avionic and stores management integration. In the future it will be used to extend the systems integration to flight controls, propulsion controls, and vehicle management systems (electrical, hydraulic, environmental controls, etc.). The following descriptions of the MIL-STD-1553 elements were extracted from Reference 16.

### 1. Elements

The key MIL-STD-1553 revision B elements are the bus controller, the embedded remote terminal (a sensor or subsystem that provides its own internal 1553 interface), the stand-alone remote terminal, and the bus monitor. Two other devices that are part of the 1553 integration; the twisted shielded pair wire data bus and the isolation couplers that are optional.

The bus controller's main function is to provide data flow control for all transmissions on the bus. In this role, the bus controller is the sole source of communication. The system uses a command/response method.

The embedded remote terminal consists of interface circuitry located inside a sensor or subsystem directly connected to the data bus. Its primary job is to perform the transfer of data in and out of the subsystem as controlled by the bus controller. This type of terminal usually does not have bus controller capability. However, if the sensor itself is fairly intelligent, it can become a candidate for the backup bus controller function. Generally, an intelligent subsystem (i.e., computer based) can become a backup bus controller if a second computer, equal in function to the primary, is unavailable. [Ref. 16]

The stand-alone remote terminal is the only device solely dedicated to the multiplex system. It is used to interface various subsystems, which are not 1553 compatible with the 1553 data bus system. Its primary function is to interface and monitor transmission in and out of these non-1553 subsystems.

The bus monitor listens to all messages and subsequently collects data from the data bus. Primary applications of this mode of operation include collection of data for on-board bulk storage or remote telemetry or use within an on-line or off-line back-up controller to observe the state and operational mode of the system and subsystems.

The fourth item is the data bus itself. The standard defines specific characteristics for the twisted pair shielded cable. Notice 2, an addendum to the MIL-STD-1553, tightens these requirements and adds a definition for connector polarity. The bus data rate is 1 Mbps using Manchester II biphase encoding. The transmission technique is half-duplex; the serial word length is 20 bits of which 16 bits are used to carry data.

The last item to be discussed is the data bus coupler unit that isolates the main bus from the terminals. MIL-STD-1553B allows two types of data bus interface techniques; direct coupling and transformer coupling. Subsystems and the 1553 bus elements are interfaced to the main data bus by interconnection buses called "stubs." These stubs are either connected directly to the main bus or interfaced via data bus couplers. The data bus couplers contain two isolation resistors (one per wire) and an isolation transformer (with a turns ratio of 1 to $\sqrt{2}$). The purpose of the data bus couplers is to prevent a short on a single stub from shorting the main data bus, causing the system to fail. The selection of the

value of the resistors, the transformers turn ratio, and the receiver impedance are such that the stub appears to the main bus as a "clean interface" (i.e., high impedance). [Ref. 16]

### 2. MIL-STD-1553 Design Implementation

One of the objectives of this thesis is to present a hardware design that will facilitate the integration of tactile technology into military aircraft. One way this design accomplishes that objective is the addition of a multifunction MIL-STD-1553 interface. Although the minimum MIL-STD-1553 interface this design requires is a remote terminal described in the previous section, the design actually incorporates hardware to support dual data bus remote terminal (RT), bus controller (BC), and bus monitor (MT) capabilities. The additional capabilities give the aircraft designers one more redundancy in the 1553 system for an extra measure of safety and functionality. Furthermore, the components used in this design are fully compatible with MIL-STD-1773 (fiber optic) transceivers, which is the follow on to the MIL-STD-1553.

#### *a. The Advanced Communication Engine Integrated 1553 Terminal*

Data Device Corporation's BU-61588 Advanced Communication Engine (ACE) terminal provides a complete integrated interface between the MPC860 and the MIL-STD-1553 bus. The ACE terminal contributes BC/RT/MT functionality in a small one square inch low-profile package. The hybrid Multi-Chip Module (MCM) integrates dual transceiver, protocol, memory management and processor interface logic, and 12K words of internal buffered RAM as shown in Figure 26. This design minimizes the "glue" logic necessary to interface the ACE terminal and microprocessors. The BU-61588 is in

compliance with not only the MIL-STD-1553A/B notice 2 protocol, but also the McAir and STANAG 3838 electrical and protocol requirements. [Ref. 17]

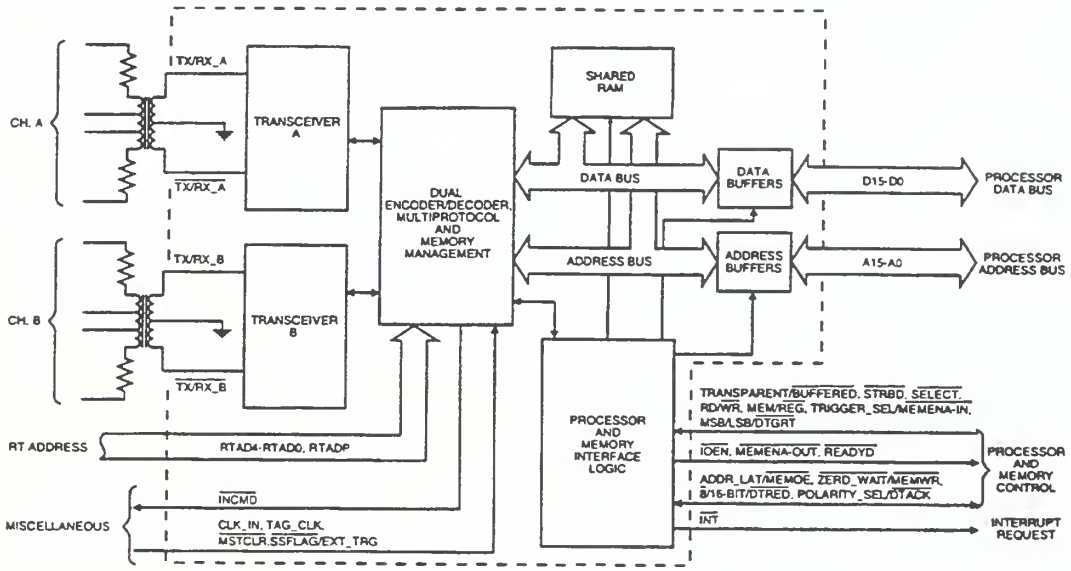



Figure 26. The ACE 1553 block diagram [Ref. 17]

The ACE terminal looks like a memory chip to the MPC860; all communications and processing for the 1553 interface is done on the ACE chip and does not burden the MPC860's CPU. Figure 51 in Appendix B shows the schematic connections for the ACE terminal. The DIP (Dual In-line Package) switches set the address of the TIMS RT, which is determined by the aircraft design engineer. Each device on the 1553 bus must have a unique address. This address may be hard-wired in the final design.

### *b. Transformers and Connector*

The MIL-STD-1553 bus specification has brought about the need for versatile pulse transformers that meet all the electrical requirements of Manchester II serial

biphase data transmission. The standard requires precise turns ratio configurations, component isolation, and common mode rejection ratio characteristics. Shown in Figure 51 are the required transformer, connector, and impedance matching resistors that couple the ACE to the 1553 bus. The hardware supports a dual-channel 1553 bus architecture, although only one channel is necessary for communication. In practice, more than one channel is needed because the bus controller will switch channels when one channel experiences a bit error rate (BER) greater than some tolerance.

## C. EXTERNAL INTERFACES

In addition to the MIL-STD-1553 interface, the microcontroller system implements three other external system interfaces. They include a serial interface, a TIC interface, and a MPC860 debug port interface. The serial interface and debug port descriptions were extracted from Reference 13.

### 1. Serial Interface

The MPC860 has four SCCs that can be configured independently to implement different protocols. Together, they can be used to implement bridging functions, routers, gateways, and interface with a wide variety of standard WANs (Wide Area Networks), LANs (Local Area Networks), and proprietary networks. The SCCs have many physical interface options such as interfacing to TDM busses, ISDN (Integrated Services Digital Network) busses, and standard modem interfaces. On the MPC860, the SCC does not include the physical interface but it is the logic which formats and manipulates the data obtained from the physical interface. The choice of protocol is independent of the choice of

physical interface. When an SCC is programmed to a certain protocol, it implements a certain level of functionality associated with that protocol. For most protocols, this corresponds to portions of the link layer (layer 2 of the seven-layer ISO model). Many functions of the SCC are common to all of the protocols. Each SCC supports a number of protocols—Ethernet, HDLC bus, BISYNC, IrDA, asynchronous or synchronous start/stop (UART), totally transparent operation, and AppleTalk/LocalTalk. Although the selected protocol usually applies to both the SCC transmitter and receiver, the SCCs have an option of running one-half of the SCC with transparent operation while the other half runs the standard protocol. Each of the internal transmit and receive clocks for each SCC can be programmed with either an external or internal source. The internal clocks originate from one of four baud-rate generators or one of four external clock pins. These clocks can be as fast as a 1:2 ratio of the system clock (12.5 MHz at 25 MHz). However, the SCC's ability to support a sustained bitstream depends on the protocol as well as other factors. [Ref. 13]

Associated with each SCC is a digital phase-locked loop (DPLL) for external clock recovery. The clock recovery options include NRZ (Not Return to Zero), NRZI (NRZ Inverted), Manchester, and Differential Manchester. The DPLL can be configured to NRZ operation to pass the clocks and data to or from the SCCs without modifying them.

Each SCC can be connected to it's own set of pins on the MPC860. Each SCC can support the standard modem interface signals (RTS, CTS, and CD) through the port C pins and the CPM interrupt controller (CPIC). Additional handshake signals can be supported with additional parallel I/O lines. The SCC block diagram is illustrated in Figure 27.

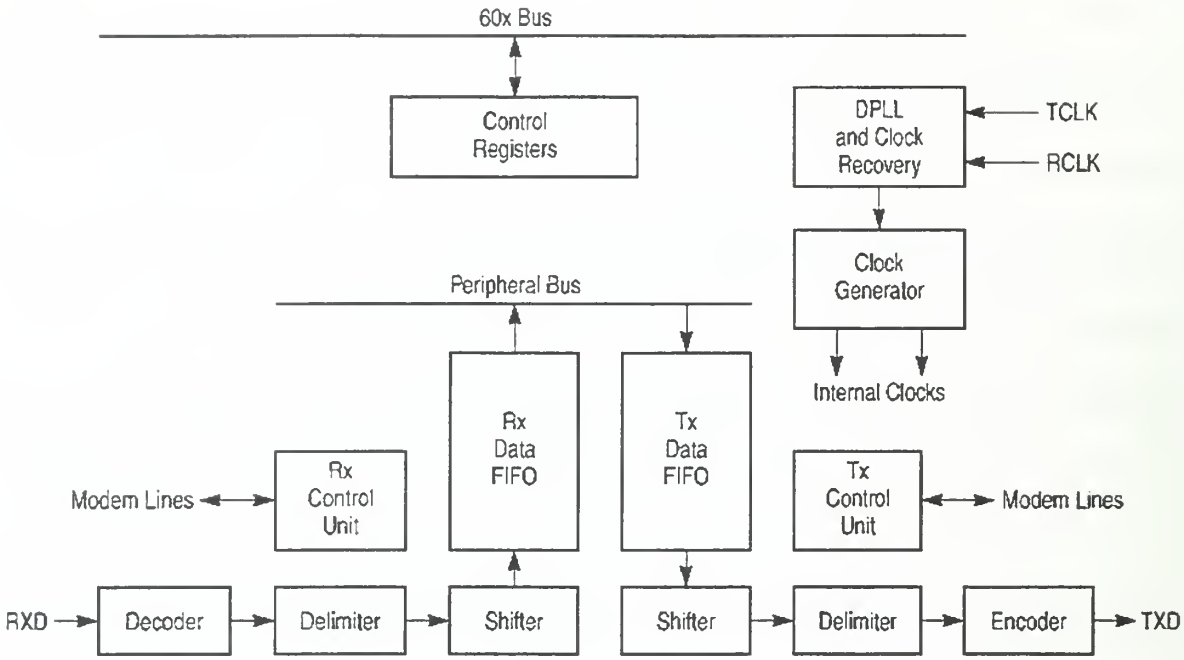



Figure 27. The SCC block diagram [Ref. 13]

Due to the architecture of the MPC860, the overall performance of the serial channels cannot be stated in absolute terms. The serial channels of the MPC860 can be programmed in many different modes, which require different degrees of processing. There are several individual bottlenecks in the system, with their own specific considerations. The maximum data rate for the SCC in UART mode assuming worst case conditions (a steady stream of minimum-size data frames) is 5 Mbps full-duplex at 50 MHz. Without serial feedback, the half-duplex maximum sustained data rate is 7 Mbps at 50 MHz.

The microcontroller serial communications circuit is shown in Figure 52. The physical interface is handled by the dual-mode RS232/422 chip. The RS232/422 driver chip has the ability to switch between RS232 protocol and RS422 protocol, making the microcontroller serial interfaces simple to change without hardware modification. The

serial port provides transmit and receive only without any handshaking. This is the configuration required by most navigation devices such as GPS.

### 2. Intelligent Tactor Interface

As detailed in Chapter III, communication to the TIC is serial using the USART protocol. The microcontroller uses one of its two serial ports to communicate with the TICs. The dual-mode RS232/422 chip drives the transmit, receive, and USART clock signals through the standard nine-pin serial connector.

The 250 Hz Carrier Wave required to drive the tactors and the counters in the TICs is not provided by the microcontroller at this time and must be generated off chip. The design of the microcontroller was completed without specifications on the power consumption needs of the tactor array and therefore it is unknown at this point whether the microcontroller system can provide enough power to drive the entire tactor array. Once the microcontroller prototype system is constructed, it should be simple to generate the required frequency from the MPC860 and to calculate the required signal amplification to drive the tactor array.

### 3. Development Port

When debugging an existing system it is sometimes helpful to be able to do so without making any changes, although in some cases it is not helpful and may even make it impossible to add load to the lines connected to the existing system. The development system interface of the MPC860 CPU supports this configuration [Ref. 13]. The development system interface of the MPC860 CPU uses the development port, which is a dedicated serial port and, therefore, does not need any of the regular system interfaces.

Controlling the activity of the system from the development port is accomplished when the CPU is in debug mode. The development port is a relatively inexpensive interface that allows the development system to operate at a lower frequency than the CPU's frequency. It is also possible to debug the CPU using monitor debugger software. Debug mode is a state where the CPU fetches all instructions from the development port. When in debug mode, data can be read from the development port and written to the development port. This allows memory and registers to be read and modified by a development tool (emulator) connected to the development port.

For protection purposes, two possible working modes are defined - debug mode enable and debug mode disable. These working modes are only selected during reset. The user can work in debug mode directly out of reset or the CPU can be programmed to enter into the debug mode as a result of a predefined sequence of events. These events can be any interrupt or exception in the CPU system, including the internal breakpoints, in combination with two levels of development port requests and one peripheral breakpoint request generated internally and externally.

The development port provides a full duplex serial interface for communications

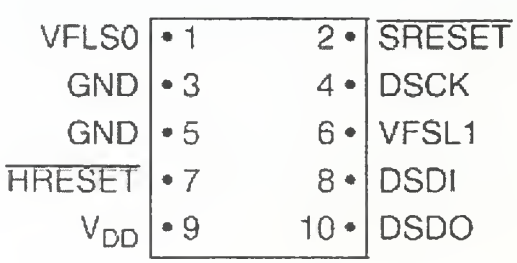



Figure 28. MPC860 debug port pinout [Ref. 13]

between the internal development support logic of the CPU and an external development tool. The following development port pin functions include serial clock (DSCK), serial data in (DSDI), serial data out (DSDO), hard reset, and soft reset (Figure 28). The schematic for the ten-pin debug port connector is located in Appendix B, Figure 59.

## D. MEMORY SUBSYSTEM

In order to meet the memory specifications for the microcontroller system, three types of memory were chosen. UV EPROM (UltraViolet Erasable Programmable Read Only Memory) was chosen as the boot memory for the system. Flash memory was chosen for high-density non-volatile program storage. SRAM (Static Random Access Memory) was added to the system for program variables and temporary data storage.

The Atmel AT22C800 UV EPROM [Ref. 18] was chosen as the system EPROM. The AT22C800 is a low-power, high performance 8-Megabit EPROM organized as either 512Kx16 or 1024Kx8 bits. It requires a single 5V power supply in normal read mode operation. For the TIMS design, two AT22C800 chips were selected with 150 ns access times. Since the EPROM is used solely as boot memory and is accessed only at startup, speed is not much of a consideration. The two chips are organized as 512Kx16 and therefore populate the entire width of the 32 bit wide data bus with 2 MB of data space. For the final design, OTP (one time programmable) TSOP (thin small outline package) chips were selected due to size constraints. For the wirewrap design prototype, UV erasable ceramic DIPs (dual inline packages) are recommended for use because of their ability to be reprogrammed after erasure. The schematic connections for the EPROM are in Figure 50.

The Sharp Electronics LH28F016LL 16 Megabit Flash memory [Ref. 19] was chosen as the system flash memory chip. Similar to the Atmel EPROM, the Sharp flash is configurable between x8 and x16 operation but only requires a 3V power supply and has a 120 ns access time. Two chips in 56-pin TSOP packages were used to populate the entire width of the system data bus, resulting in 4 Mbytes of data space. The schematic connections for the flash memory are in Figure 49.

Two Electronic Designs Incorporated (EDI) EDI8L32512C 16 Megabit SRAM memory chips [Ref. 20] are used as the temporary program storage data space. Shipping in small (approximately 1 sq. in.) 68-pin packages, these chips are among the most densely packed memory chips on the market at this time. The memory access system is organized as a 512Kx32 bit array which matches quite nicely with the 32 bit data bus. Two chips are used in the TIMS design, resulting in 4 Mbytes of SRAM. The EDI SRAM is available in 3V and 5V versions and with access times of 15, 20, and 25 ns. The TIMS design assumes 3V power supply and 25 ns access time. The schematic connections for the SRAM are in Figure 48.

## E. CLOCKS AND POWER CONTROL

### 1. Clock Circuit

A key decision concerns the type of source to be used for the system clock. The choice generally lies between the following methods [Ref. 21]:

1. A purpose-designed Hybrid Crystal Clock Oscillator, the type normally found in a metal can with four pins and similar to a DIP.
2. A suitably available and stable on-board clock waveform from another section of the total design.
3. A self-oscillating crystal oscillator composed of a crystal, with tuning capacitors and supplementary resistors.

Most designs will move faster through the design, debugging and production phases if the choice is made from methods 1 or 2. Generally, method 1 is used because no comparable alternative exists on the board already. If method 2 is selected, it is important that it be a jitter-free source with extremely low noise constituents, since any shortcoming at this point in the circuit may produce subsequent problems in other parts of the chip, such as the Phase Lock Loops.

The Hybrid Crystal Clock Oscillator used in method 1 is generally a unit that is well designed by experts in their field. Therefore, it may be relied upon unfailingly to start at low rail voltage and to perform in a guaranteed manner over a full temperature range. All specifications including jitter and noise components are known and controlled. The benefit of these qualities cannot be overstated in the total system behavior since the overall performance is critically dependent on the quality of the clock source. Often a double frequency clock is generated and is followed by a divide-by-two circuit. This produces a high amplitude voltage swing with accurate matching of each half of the clock waveform to be input to the EXTAL pin. This method was not used in the TIMS design because it was not present in the MPC860 ADS design [Ref. 11].

It is important in such circuits that no connection is made to the XTAL pin, even for measurement. Clock output pins (CLKO) are provided on the MPC860 for any external

connections. It is an advantage of this type of circuit that it will be unaffected by typical changes in chip parameters when revisions and shrinks in size are made over the lifetime of the part. These important advantages are not shared by method 3. The schematic for the microcontroller clock circuit is shown in Figure 57. The clock is a 32.768 kHz crystal oscillator, which equates to a 50 MHz MPC860 internal clock speed. The clock method chosen is identical to that in the MPC860 ADS design [Ref. 11]. The choice of crystal oscillator was made due to the lower power requirements and minimal size of this type of clock generation. Additionally, the crystal oscillator is critical for real-time operation. In testing and evaluation of the design it may be deemed necessary to add a 50 or 66 MHz canned oscillator for clock system stability.

### 2. Power Supply Circuit

The power supply circuit, shown in Figure 53, contains five components. The first is the power supply connection jumper. This is, essentially, the on/off switch of the non-battery power supply. The power connection requires a clean five-volt power supply. The second component is the DC-to-DC converter that takes the five-volt supply and converts it to a three-volt power supply. Most of the components in TIMS run off three volts. The third component is a set of light emitting diodes (LEDs) which are power-on indicators for the five-volt and three-volt supplies. The fourth component is the battery power connection. If there is backup battery power available (also referred to as keep-alive power), it will be connected to the KAPWR jumper. See the power section in the MPC860 overview earlier in this chapter for details. The last component is the power supervisor. It

monitors the three-volt power supply and switches to battery backup power if it detects a drop in the main power supply. It also initiates a power-on reset if necessary.

## F. RESET CIRCUITRY

The MPC860 has several inputs to the reset logic:

1. Power-on reset (PORESET)
2. External hard reset (HRESET)
3. System reset pin (SRESET)
4. Loss of lock
5. Software watchdog reset
6. Checkstop reset
7. Debug port hard reset
8. Debug port soft reset
9. JTAG reset

All of these reset sources are fed into the reset controller and, depending on the source of the reset, different actions are taken. The reset status register (RSR) reflects the last source to cause a reset. The reset block has a reset control logic that determines the cause of reset, synchronizes it if necessary, and resets the appropriate logic modules. The memory controller, system protection logic, interrupt controller, and parallel I/O pins are initialized only on hard reset. Soft reset initializes the internal logic while maintaining the system configuration. When a hard reset event occurs, the MPC860 reconfigures the hardware system as well as the development port configuration. The logical value of the bits that determine the initial mode of operation are sampled either from the data bus or from an internal default constant. At the sampling time the external RSTCONF pin is asserted and the configuration is sampled from the data bus, otherwise, it is sampled from the internal default.

The microcontroller reset circuit, Figure 54, contains three push-button reset buttons to initiate a hard reset, soft reset, or program interrupt. The eight DIP switches set the logic value of the bits that determine the MPC860 initial mode of operation after a hard reset.

## G. OTHER CIRCUITRY

### 1. Level Transceivers and Buffers

The level transceivers take the three-volt signals from the MPC860 and increase the voltage to five-volts on the data and signal buses for those components that require five-volt signals such as the EPROM and MIL-STD-1553 ACE. The level transceivers have two-way operation and translate the voltage back to three-volts when data flows in the opposite direction. Figure 55 presents the schematics for the transceiver circuit.

The buffers perform the same operation as the level transceivers except they have one-way operation only and operate on the address and signal buses. Figure 56 provides schematic details of the buffer circuitry. The rationale for using buffers and level transceivers comes from the examples given in the MPC860 ADS schematics [Ref. 11]. It is possible the level transceivers and/or the buffers may not be necessary, especially if no off-board memory will be used. This will have to be borne out in the testing and evaluation phase of the design process.

### 2. Pull-up Resistors

On the MPC860, unused inputs should be tied high or low, but not left floating. Unused inputs may be tied directly to VCC or GND or through pull-up or pull-down resistors to VCC or GND. For most pins, VCC can be either 3.3V or 5.0V. Unused output

pins may be left unconnected. The parallel port pins on the MPC860 do not have internal pull-up or pull-down resistors. Unused parallel I/O pins may be configured as outputs after reset and left unconnected [Ref. 22]. The schematic for the pull-up resistors is found in Appendix B, Figure 58.

# V. MICROCONTROLLER SYSTEM PRINTED CIRCUIT BOARD DESIGN

This chapter is designed for the use of PCB engineers who desire to reproduce or improve the TIMS PCB layout. It describes the methodology of component selection, specifics of the software design tools, and the board layout and etch methodology.

## A. PCB COMPONENT SELECTION

A critical step in the PCB design process is component selection. Ideally, this step is performed before the schematic design phase. Five questions must be answered when selecting components:

1. Does the part perform the desired function?
2. Does the part meet timing, voltage, and power requirements without additional hardware?
3. Does the part fit compactly into the PCB design?
4. Does the part meet budget requirements?
5. Does the part meet reliability specifications?

When selecting components and answering the previous five questions with affirmatives, it becomes clear that question two is often in conflict with question three, and question four often conflicts with all of the others. Furthermore, each fiscal quarter new components are produced that are faster, smaller, and use less power than the ones in the previous quarter. Thus, by the time this design is published, the components used in this prototype TIMS design may no longer be the best available. At this writing, however, they are the most up-to-date COTS parts available.

The choice of the MPC860 as the central controller of TIMS was the most difficult and challenging decision of the entire design process. Many microprocessors and microcontrollers were considered and rejected. Stand-alone microprocessors offered better performance than integrated microcontrollers, but the additional "glue" logic required to add functionality to the microprocessors took up too much PCB space and power. Other microcontrollers were considered but none had the copious built-in features and communications capability of the MPC860. The TIMS system is, in its most basic form, an intelligent communications "hub" between the aircraft's sensors and the tactor array. Once the MPC860 was chosen as the system microcontroller, the peripheral chips were simpler to identify.

Another key design decision was to populate the entire width of the MPC860's 32-bit data bus. This meant using 32-bit wide memory chips, which can be difficult to find. Only 32-bit wide SRAM chips were found, so the Flash and UV EPROM memory systems each required two 16-bit wide chips to populate the full width of the data bus.

The choice of the MIL-STD-1553 ACE and transformer components has already been discussed in the previous chapter, as has the UART serial interface chip and the clock oscillator chip. It is worth repeating here that the RS232/422 serial interface chip can drive up to two serial ports in either RS232 or RS422 protocols, and adding additional UART serial ports is simply a matter of adding one more RS232/422 serial interface chip and connectors.

One of the critical choices made when searching for and choosing components for the TIMS design was the choice of chip package. Most integrated circuits (ICs) come in

more than one package. Some examples of standard packages include DIP, PGA (pin grid array), QFP (quad flat pack), PBGA (plastic ball grid array), and TSOP (thin small outline package). Integrated Circuit (IC) packages can be separated into two main groups: surface mount devices and through-hole devices. Through-hole devices refer to the fact that the pins on the IC package connect through each layer of the PCB, while surface mount devices connect only to the first layer of the PCB. When designing a multi-level PCB, surface mount devices provide extra flexibility in connecting the devices to each other and placing the components on the board. Conversely, through-hole devices restrict the designer on both connection strategies and device placement. However, through-hole devices are more reliable in high vibration and high-G applications, such as in a jet aircraft or a helicopter. Also, PCB assembly is usually easier and less expensive when through-hole devices are used. These concepts will be discussed in more detail in the following sections of this chapter.

## B. CADENCE DESIGN SOFTWARE

Various programs from the Cadence family of design software were used to generate all of the schematics and PCB layouts in this document. The two most prominent software programs were the Concept schematic capture program and the Allegro PCB design program. This section assumes a familiarity with the Concept schematic package and focuses on PCB design using the Allegro software package. For a more in-depth overview of the design process, refer to the Concept and Allegro online documentation.

Allegro is a physical layout system for PCB design. With Allegro, a designer can place and route a design, as well as generate the output and documentation necessary for the manufacture of the design. The Allegro PCB design process has the following general steps:

1. Library Development
2. Logic design creation and transfer
3. Layout Preparation
4. Design Layout (placement and routing)
5. Optional Design Analysis (thermal, signal, electromagnetic, and reliability)
6. Design Completion
7. Generating manufacturing output

During the Library Development step, the components for the schematic design are selected. Each component must have a physical package associated with the component. Different types of surface mount and through-hole packages (QFP, PBGA, PGA) were introduced in the previous section. If the components or their associated physical packages, which Cadence refers to as "symbols," do no exist in the Cadence design libraries, they have to be created by the designer. Physical packages can be created in the Allegro Symbol Editor, and components can be created and associated with symbols in the Cadence RapidPart program. Furthermore, when creating symbols, the designer must provide contact points around each symbol so the symbol pins may contact the PCB. These contact points are called "padstacks" and must be edited separately from and then linked into the symbols. Once all components, symbols, and padstacks are created, the designer must then create the schematic in Concept. When complete, the design must compile with no errors.

The PCB layout preparation can be performed in parallel to the Concept schematic design. The layout preparation takes place in Allegro and includes drawing the outline of

the PCB, defining the layers of the board (cross section), and defining the PCB design rules such as minimum etch width and minimum component separation distance. Multiple layer boards provide the designer with additional flexibility but are more difficult to design and cost more to manufacture. After the number and composition of the PCB layers has been decided, the connections between the layers (vias) must be created using the padstack editor. Through vias connect the top layer to the bottom layer, buried vias connect either the top or bottom layer to a middle layer, and blind vias connect two middle layers. Allegro has a program that will automatically create all through, blind, and buried vias for a multi-layer design given an example via.

When the layout preparation and schematic design are complete, the schematic data must be associated with the PCB layout. The Physical Design Flow menus in Allegro provide a step-by-step process for transferring the schematic data from Concept to Allegro. Once the schematic data is transferred to Allegro, the Concept schematic can be "back annotated." This process automatically annotates the entire Concept schematic with part identifiers and pin numbers, which can then be cross-referenced with the PCB design.

The Design Layout process involves placing the packaged components on the PCB and routing thin metal wire called 'etch' to interconnect each component such that the PCB connections match the schematic connections. It is highly recommended that each layer of a multi-layer board be assigned a separate color, and all padstacks, etch, and vias separate layers maintain a constant color scheme. Allegro has the capability to automatically place components and route etch. It is not recommended to use these automatic features since manual placement and routing invariably produces superior results. At this point, it is

useful to reference a good design rule guide such as the Motorola MPC860 Design Checklist [Ref. 22] or another PCB guide. A PCB guide contains information such as minimum etch width and etch spacing, minimum component spacing, minimum etch-component spacing, etc.

After completing the component placement and routing, Allegro has programs that check the design for design rule violations and perform reliability analysis. Additional programs perform thermal analysis, signal integrity analysis, and electromagnetic interference analysis. These Design Analysis programs work only if the information about the ICs and their corresponding packages is available at design time. Some of these programs will catch errors earlier if run after component placement and before etch routing.

During the Design Completion and Manufacturing Output steps, power and ground planes are added, artwork (copyright dates and version numbers) is added to the PCB, and manufacturing diagrams and files are created.

## C. PCB COMPONENT LAYOUT AND INTERCONNECTION

Appendix C provides PCB layout diagrams for this design. The PCB diagrams are to scale; the board is roughly 3.375x7.0 inches. The first diagram in Appendix C, Figure 60, shows the PCB layout with all components present and all interconnection etch. Figure 61 shows the same PCB with the interconnection etch removed and each part labeled with a part identifier, often called a reference designator. The reference designator is cross-referenced to the parts list in Appendix D and the reference designators in the schematics in Appendix B. All components shown in the diagrams are mounted on the top layer of the

PCB. It is possible to mount some of the surface-mount devices on both the top and bottom layers of the PCB. This would further shrink the cross-sectional area of the PCB design. From rudimentary calculations, is seems possible to shrink the design to almost half its present size and is an exercise left to future TIMS PCB designers.

The PCB board contains seven etch layers separated by non-conducting dielectric. They are, in order from top to bottom,

1. Layer Top (TOP)
2. Signal Layer Top (SIGTOP)
3. Ground Layer (GROUND)
4. Signal Layer Middle (SIGMID)
5. Power Layer (POWER)
6. Signal Layer Bottom (SIGBOT)
7. Layer Bottom (BOTTOM)

Through, blind, and buried vias connect the different layers to one another. Great care was taken to follow standard PCB design practices. For instance, ensuring no interconnect was too long or too thin and that all etch lines bent at angles 90 degrees or greater.

Figure 62 - Figure 68 divide the PCB design up into seven sections, which approximately correspond to the schematic divisions presented in Appendix B. Of particular interest in Figure 62 are the resistor packs. When several resistors of the same value have a common terminal, it is a space-saving technique to use resistor packs instead of multiple independent resistors. Also of note is Figure 64, which contains the layout of the MIL-STD-1553 components. This board was designed so the TIMS prototype could interface with modern aircraft sensors. Clearly, these components take up almost one-third of the PCB area. It is therefore very advantageous to remove these components when

designing a TIMS prototype for those applications in Chapter II which do not require a 1553 interface.

# VI. RESULTS, CONCLUSIONS, AND RECOMMENDATIONS

TIC logic design simulation was accomplished using the design software and the results are presented in the following section. Construction and testing of the TIMS PCB prototype was not possible due to an unfortunate cut in project funding early in the design cycle. Suggestions for future work are presented at the end of this chapter.

## A. TACTOR INTERFACE CHIP PERFORMANCE RESULTS

The test procedure for the TIC design was performed with Cadence OpenSim software and the outputs were viewed with Cadence Waveview package. A list of the test procedure follows:

1. Perform an adequate test on EI/RP combinations for correct Tactor Driver module output.
2. Test constant tactor on and constant tactor off features
3. Address the tactor with a correct address, an incorrect address, and the broadcast address
4. Test each of the error conditions (parity error, frame error, sequence error)
5. Test single commands (Reset, Transmit Feedback)

The reason the test for EI/RP combinations must be simply adequate and not exhaustive is that the logic blocks that generates the EI and RP signals are independent. Therefore, the $2^{12}$ valid EI/RP combinations for which EI is less than RP do not need to be tested individually to ensure correct operation. Instead, each of the EI values ($2^6$) have to be tested with only one RP, and each of the RP values have to be tested with just one EI to ensure correct TIC operation. The result is only $2^7$ tests vice $2^{12}$. Additionally, the design for the Tactor Driver module is algorithmic and depends solely on the values loaded into the

counters. Consequently, if the Tactor Driver module produces correct results for a few representative EI/RP combinations then the rest of the combinations will produce correct results as well. Similar to a linear time-invariant system, the Tactor Driver module output is shifted and scaled by the EI/RP input, but the shape of the output does not change. The EI/RP combinations tested are listed below (all values are decimal and are in milliseconds).

1. 0/x (constant off), where x represents a don't care
2. x/0 (constant on)
3. 16/64, 32/128, 64/256, 128/512, 256/1024

The tests were performed with the USART clock running at 1, 5, and 10 MHz, and the Tactor Driver clock running at 250 Hz and 1 MHz. All test results were satisfactory. All modules performed as expected and the correct operation was verified during each test. During the tests when the USART clock was running at 10 MHz, the gate delay of the TTL logic gates was starting to affect the circuit adversely. In accordance with these results, it is recommended that a TIC design implemented with TTL devices run at a clock speed of less than 10 MHz.

## B. CONCLUSIONS

SD mishaps are depleting aviation communities of manpower and material resources. TSAS is one feasible and successful solution for improving SA and decreasing SD mishaps in aviation environments as well as improving performance in VSWMCM operations. However, TSAS is a research system and, therefore, needs to be modified to be compatible with tactical environments. TIMS, using the Intelligent Tactor concept, provides a viable and tactically compatible system design that can finally put tactile

technology into the hands of the warfighter. The design of TIMS presented in this document exceeds the specifications detailed in Chapter III and incorporates advanced features that will improve the capabilities and performance of the current TSAS prototype. The strength of TIMS is evident in the ability of the system to communicate with multiple sensor and information systems (i.e., the MIL-STD-1553 bus and GPS). Because of this ability of TIMS to interface with many systems, TIMS can enable these systems to be used to a fuller extent by improving the man-machine interface. Few systems have such potential to provide performance improvement in such a wide range of applications. Clearly, the time has come for tactile technology to supplement the tools of the 21st century warfighter. The success of TSAS has proven the concept of tactile information presentation, and it is now time for TIMS to become the means that will carry the technology into the field and into the cockpit.

## C. SUGGESTIONS FOR FUTURE DEVELOPMENT

While the construction of the TIMS prototype has yet to be accomplished at the NPS, the designs of the TIC and the microcontroller system are, for the most part, completed. With additional funding, it will be possible to construct the TIMS prototype and provide the warfighter with an exciting new technology.

### 1. TIMS Prototype Construction

The TIMS prototype construction procedure involves more than simply submitting the design presented in this thesis to a PCB manufacturer. The design presented herein relies on information promulgated by component manufacturers, which is not always

reliable. The design must first be verified to ensure correct interaction of each component. Therefore, it is recommend that the Motorola MPC860 Application Development System be used to test and evaluate the TIMS system before PCB prototype construction.

The MPC860ADS is an Applications Development System designed to aid hardware and software developers of the MPC860 PowerQUICC in quickly evaluating and developing applications for this device. All of the hardware resources needed to download and debug application software are provided, such as large blocks of flash and DRAM for the processor, serial port, clock generation option circuitry, logic analyzer connectors, expansion connectors as well as monitor/debugger hardware and software. The logic analyzer connectors provide the user with access to all of the processor's pins in order to monitor bus activity. The expansion connectors let the user attach user hardware applications and utilize board resources to verify a design. This board could also be used as a demonstration tool. The board may be disconnected from the host computer after application software has been burned into its flash memory. The portable board may then be connected to a suit of tactors to demonstrate tactile technology at conferences or exhibitions.

To serve as a convenient platform for software development, the MPC860ADS is provided with MPC8Bug, a monitor/debugger, for the PowerQUICC. The monitor/debugger provides operations of memory dump and set, single instruction execution, breakpoints and downloads, as well as a self-test verification suite. This monitor is downloaded via the MPC860ADI-PC or MPC860ADI-SUN4, which plugs into the user

host system and connected to the ADS via a parallel ribbon cable. The MPC860ADS board has a Ball Grid Array (BGA) socket to accommodate the MPC860 and its successors.

The TIMS microcontroller design may be verified by wire-wrapping the memory chips, 1553 interface chips, and other components to project boards and connecting the project boards to the expansion connectors on the MPC860ADS. It is useful to use sockets for the memory and other chips on the project boards so the chips may be removed and replaced easily during testing and evaluation. Once the TIMS design herein has been verified as much as practical in hardware, it may be submitted to a PCB manufacturer for component placement and etching.

### 2. Tactor Interface Chip Implementation

As noted in Chapter III, the hardware implementation of the TIC has yet to be provided. It would be useful to implement the logic in an FPGA or CPLD first to test the logic design, then design and fabricate a single-chip VLSI solution. Furthermore, some TIC features have not, as yet, been implemented in the design presented in Chapter III. Two examples are the ability of the TIC to adjust the voltage of the tactor waveform and the ability of the TIC to vary the frequency of the 250 Hz Carrier Wave.

### 3. Software Development

Except for the specifications listed in Chapter II, there has been no mention of software issues in this document. There are many software companies that offer real-time operating systems that have support for C++ and ADA. Two of the largest are Green Hills Software and Wind River Systems. The Green Hills Multi software development system is

ideal for the MPC860 TIMS applications. The operating system is real-time, scalable, and has a wealth of hardware and software support for the MPC860 on multiple development platforms. Using the MPC860ADS and the Multi software design environment, hardware and software development can take place simultaneously, greatly reducing design cycle time.

# APPENDIX A. INTELLIGENT TACTOR SCHEMATICS

The diagrams in this appendix are the design schematics for the TIC. The schematics are listed below.

1. Tactor Interface Chip Logic
2. Serial Data Shift Register
3. Serial Data Protocol Check
4. Data Decode
5. Command Processor
6. Auxiliary Command Processor
7. Serial Feedback Transmitter
8. Tactor Driver
9. State Machine
10. Odd Parity Check and Generator
11. 8-Bit Latch
12. 4-Bit Synchronous Up Counter
13. 6-Bit Synchronous Up Counter
14. 4-Bit Synchronous Down Counter
15. 6-Bit Synchronous Down Counter
16. 16 x 1-Bit Multiplexer
17. Edge-Triggered D Flip-Flop
18. Edge-Triggered T Flip-Flop

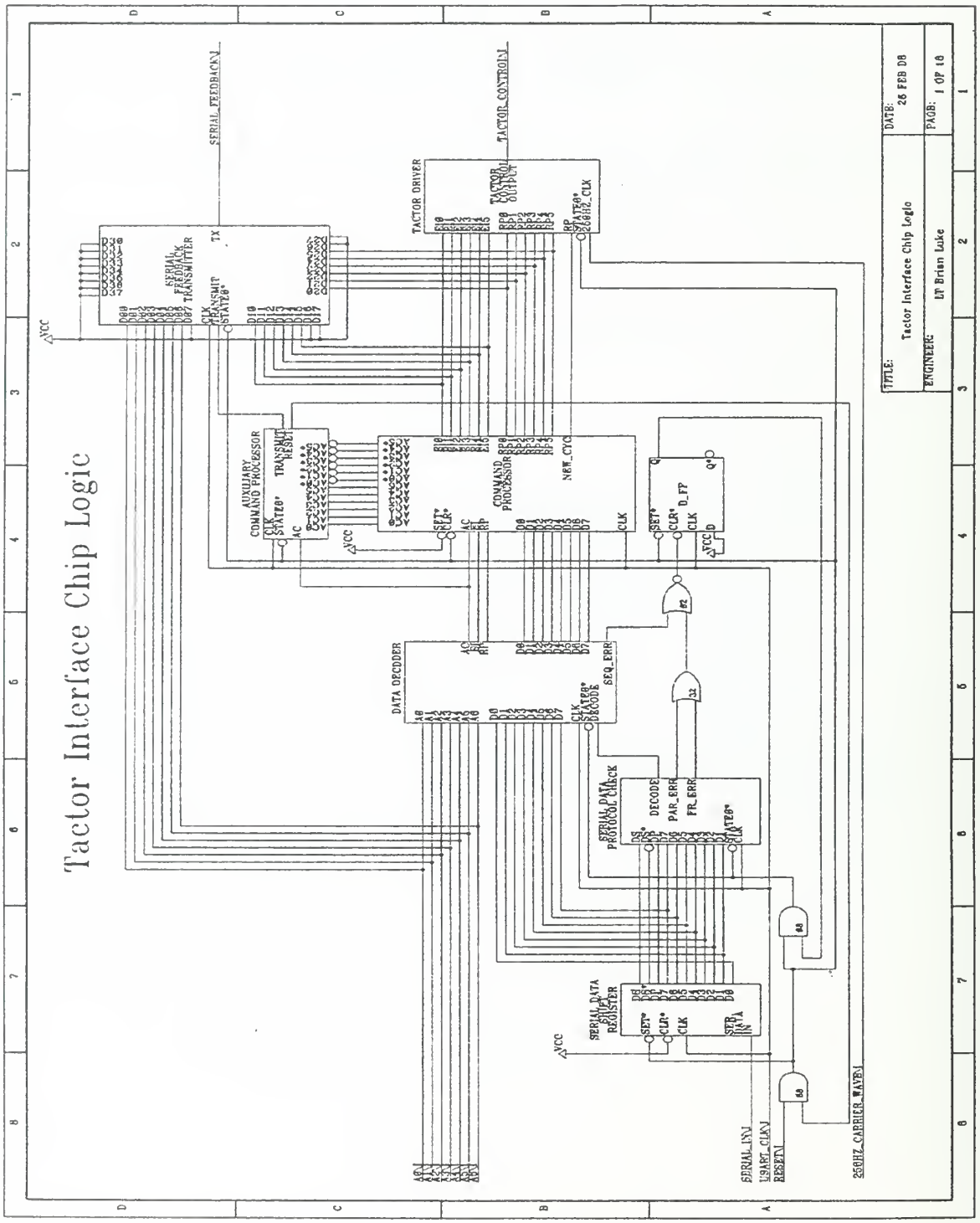



Figure 29. Tactor Interface Chip logic

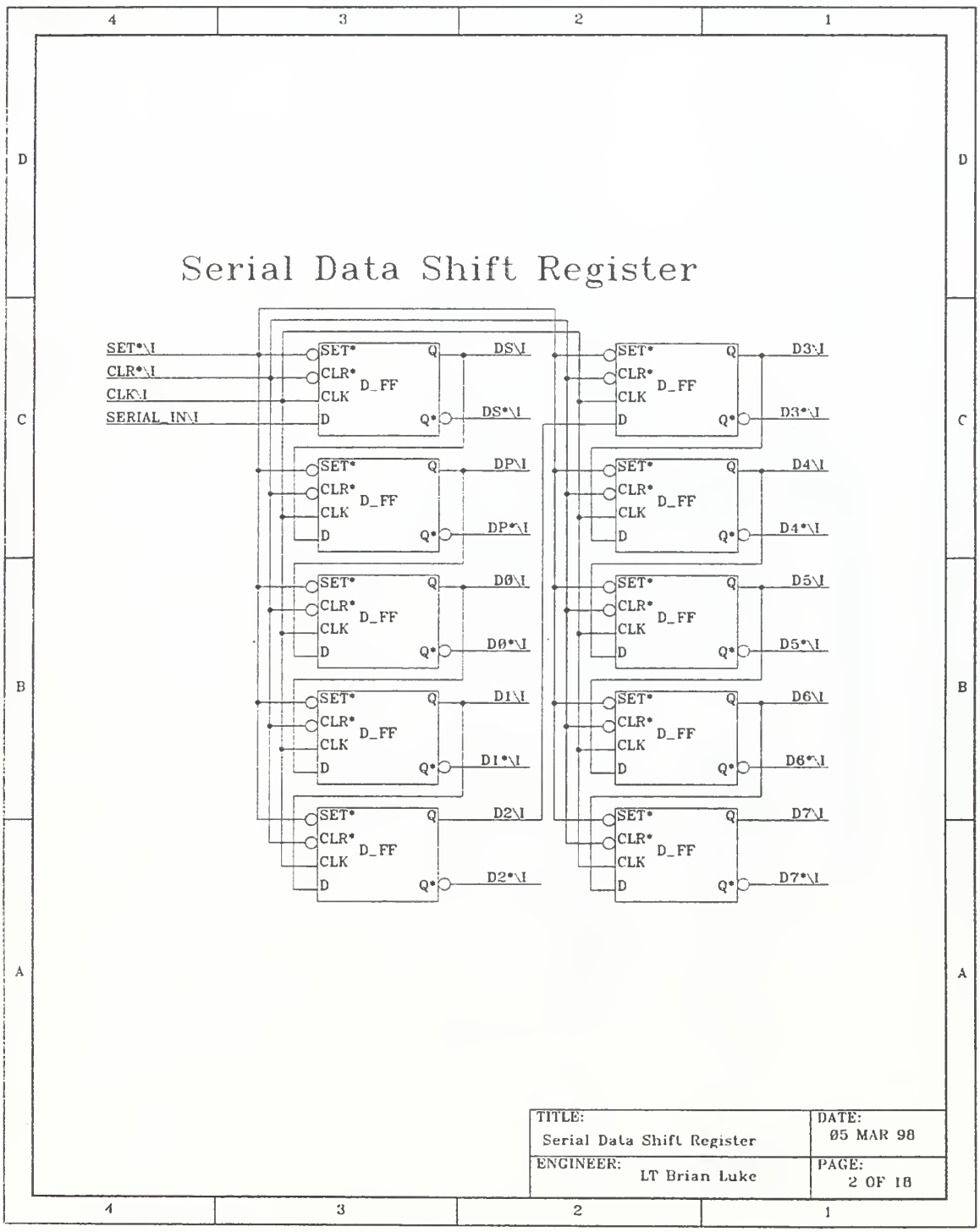



Figure 30. Serial Data Shift Register

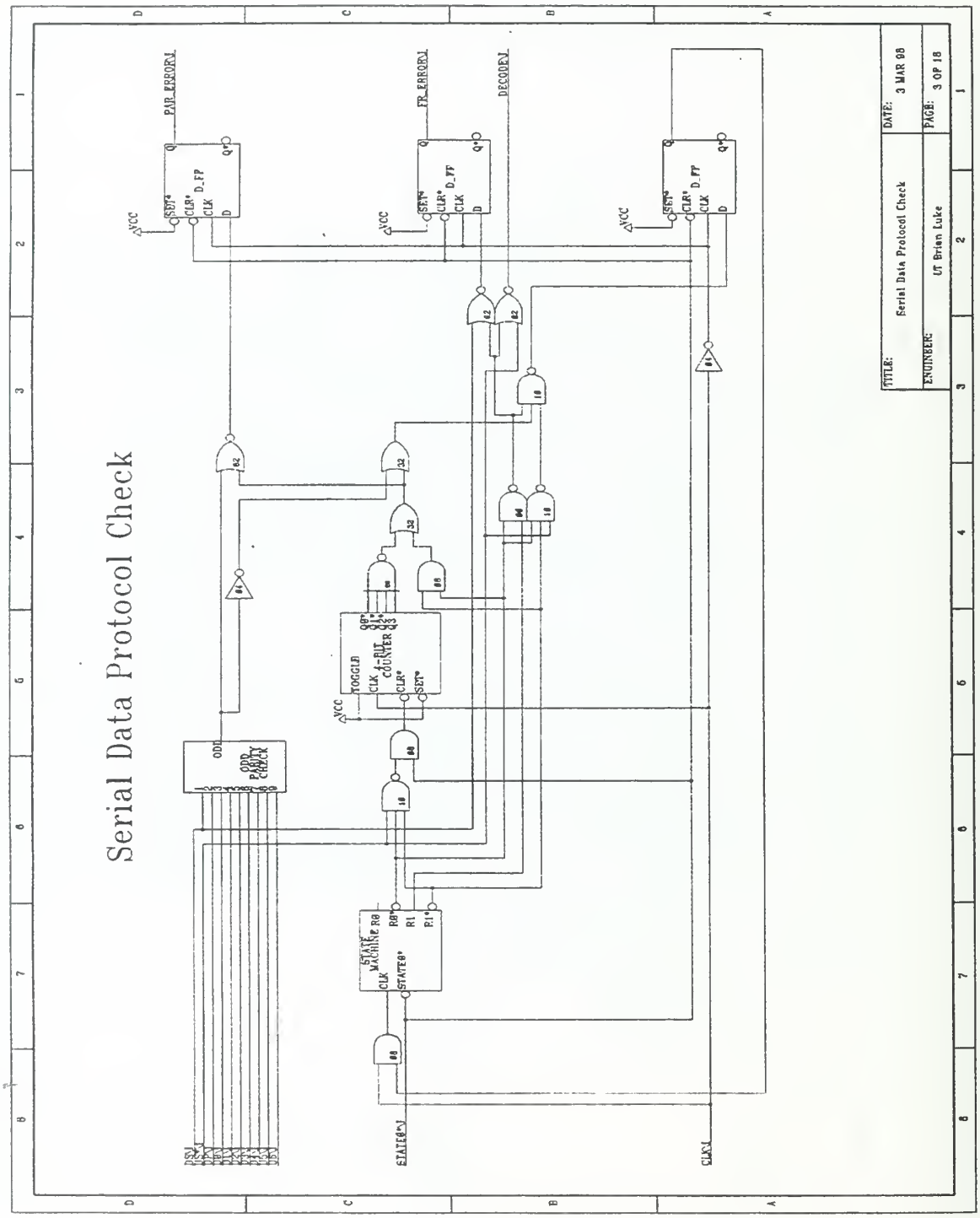



Figure 31. Serial Data Protocol Check

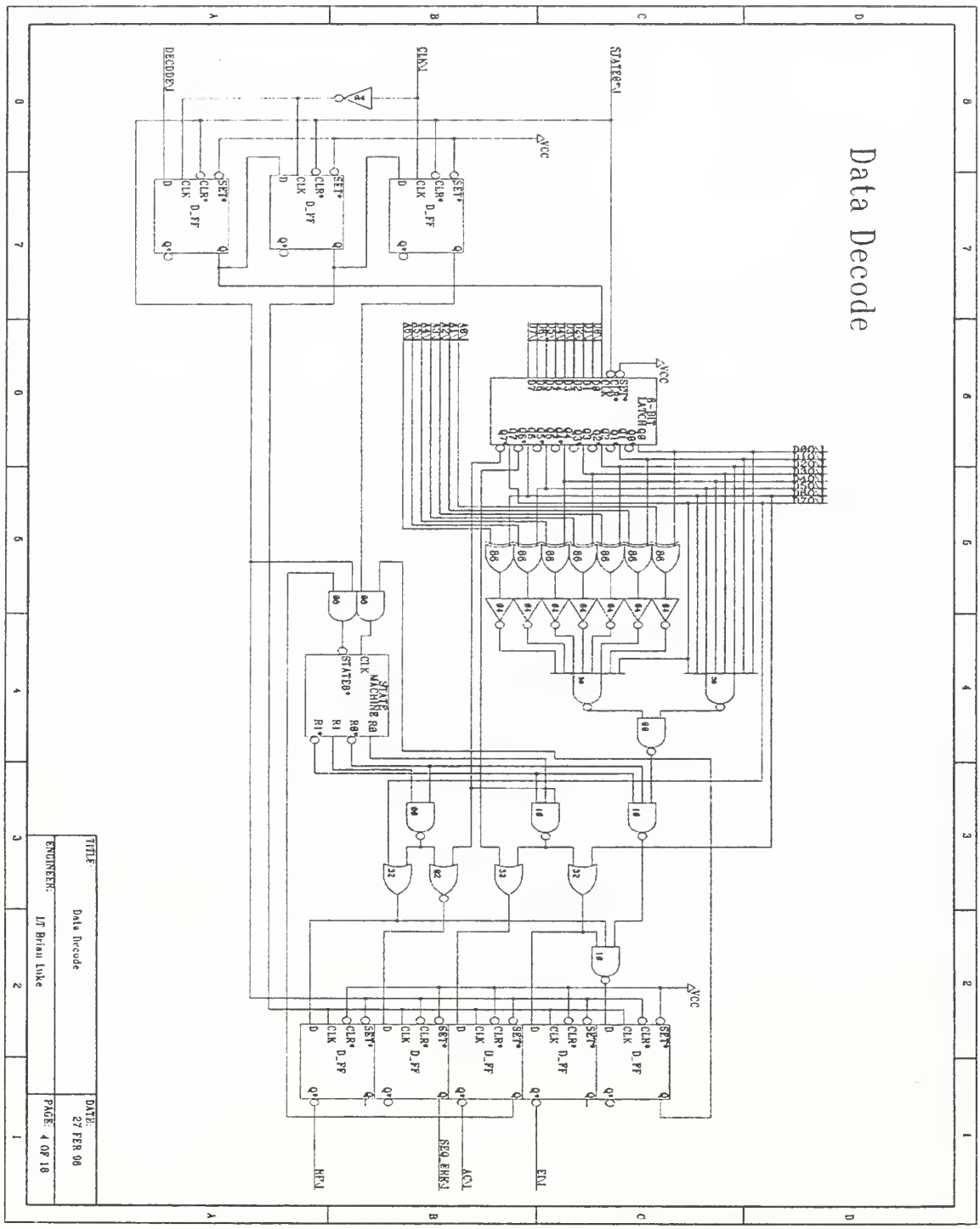



Figure 32. Data Decode

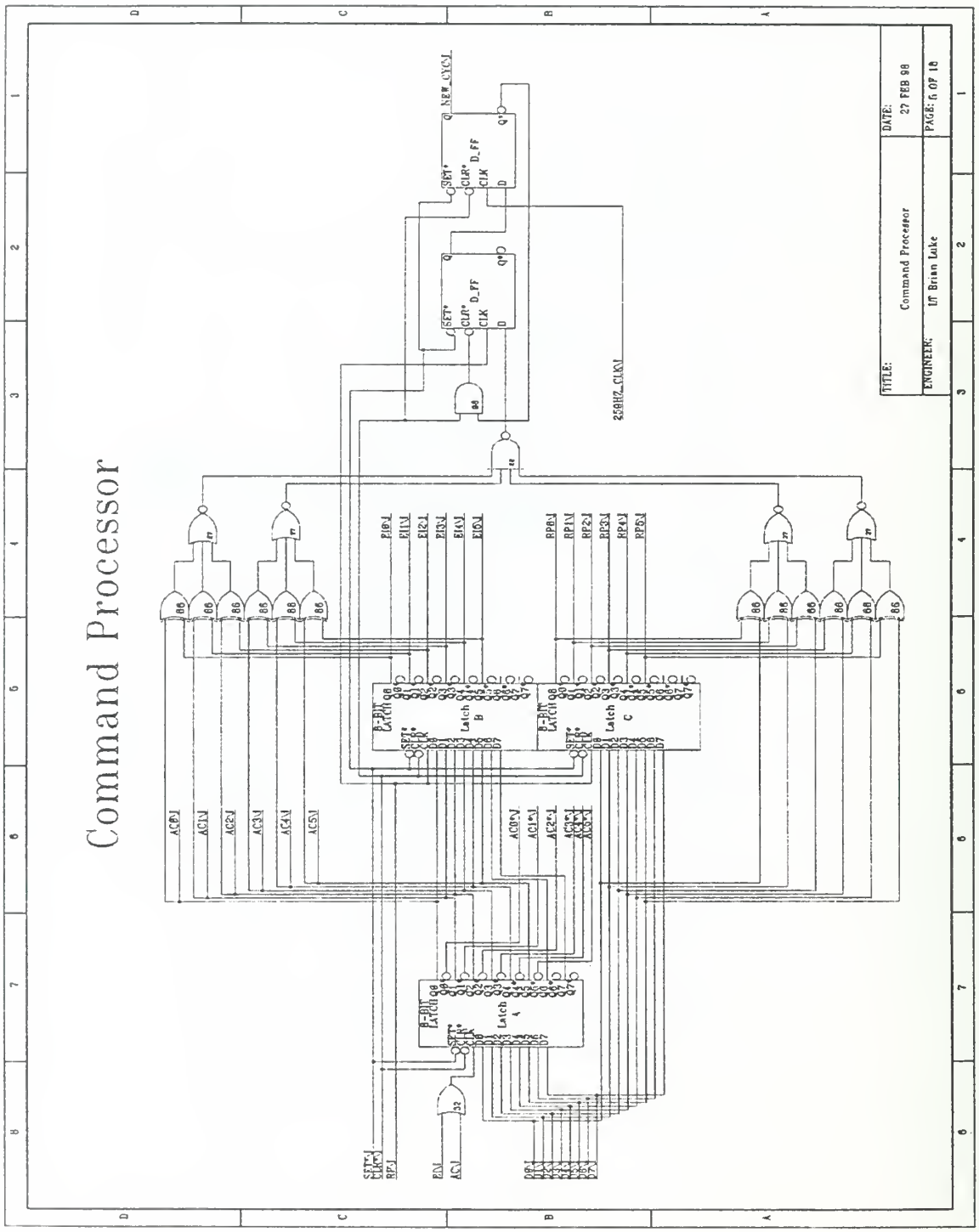



Figure 33. Command Processor

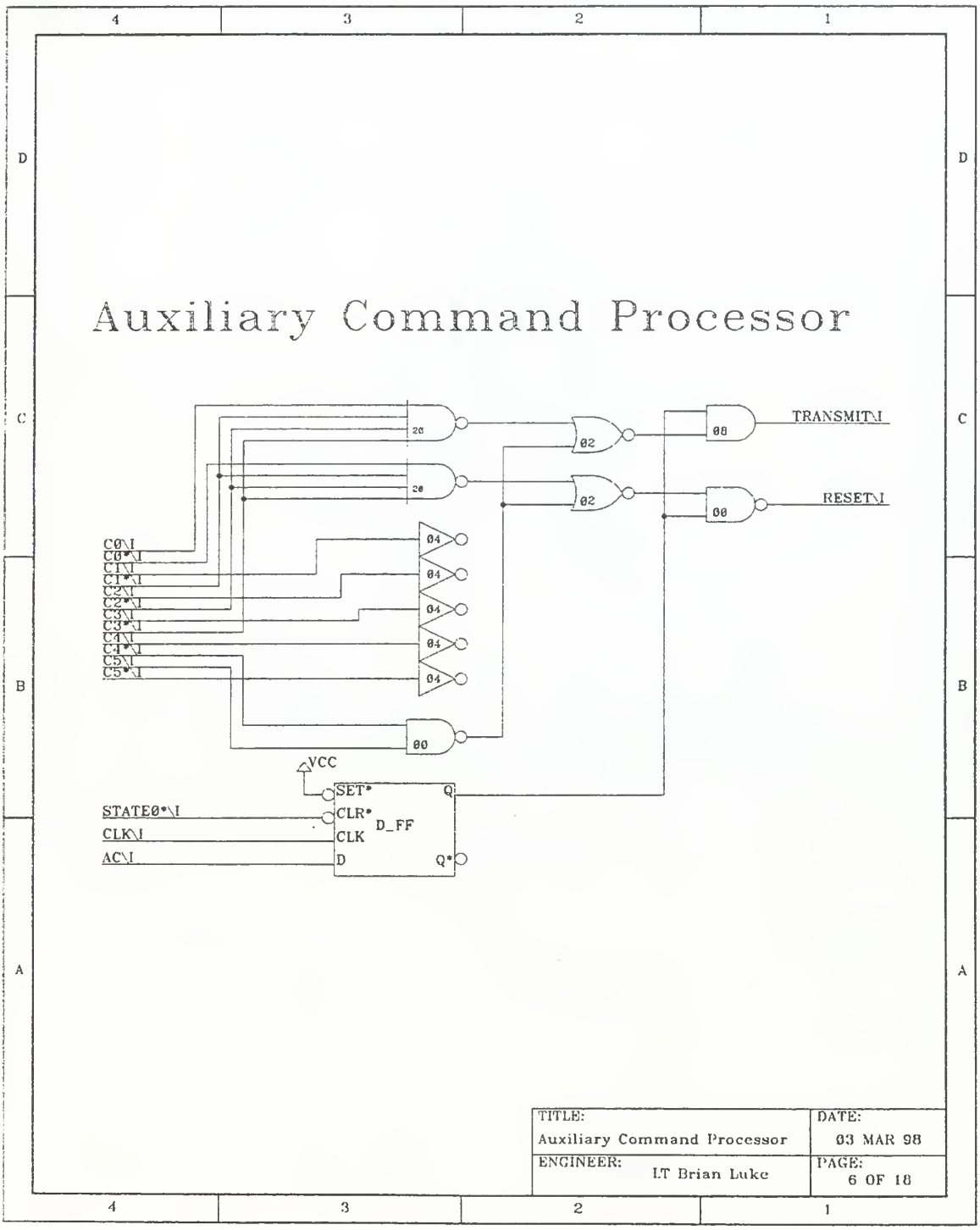



Figure 34. Auxiliary Command Processor

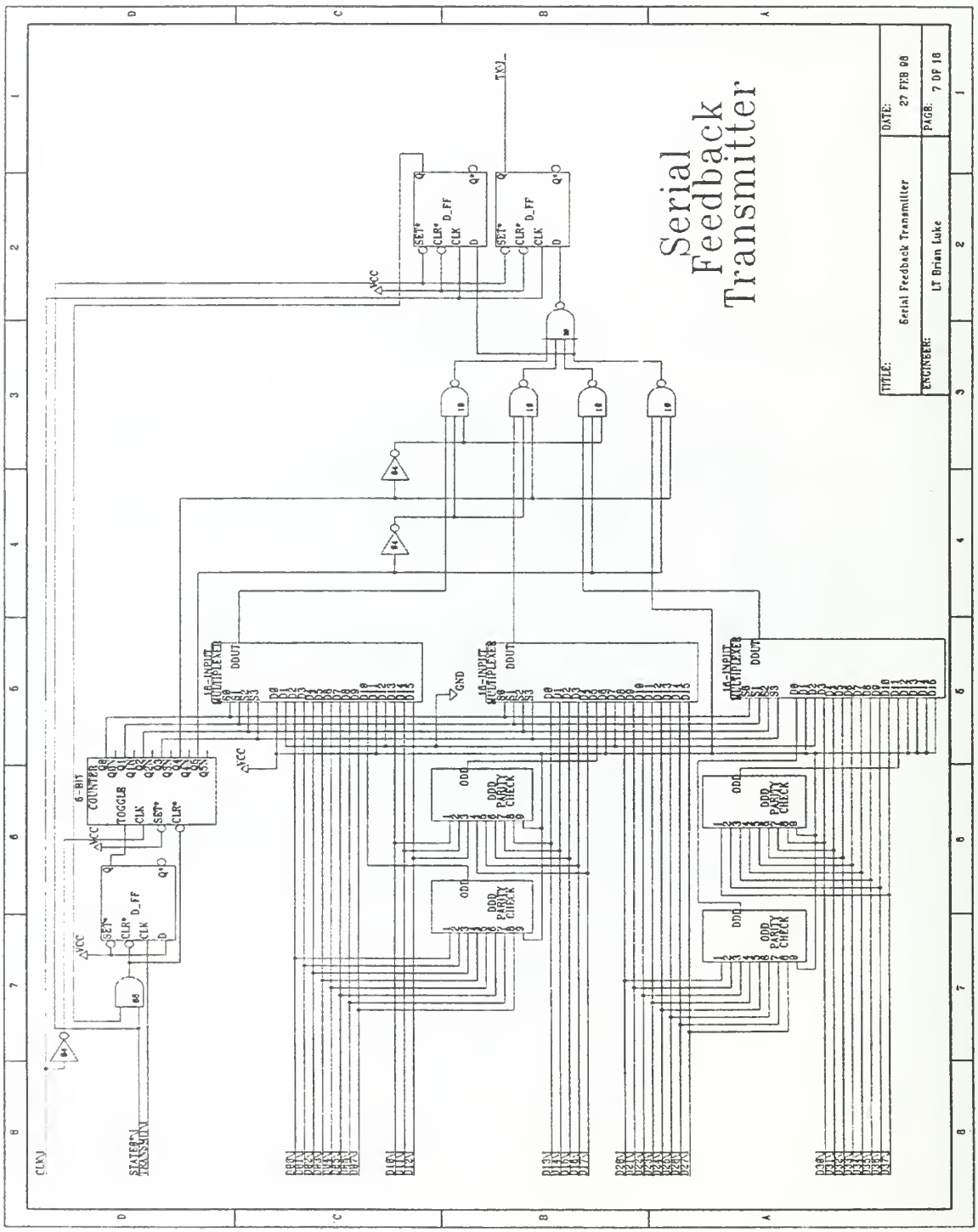



Figure 35. Serial Feedback Transmitter

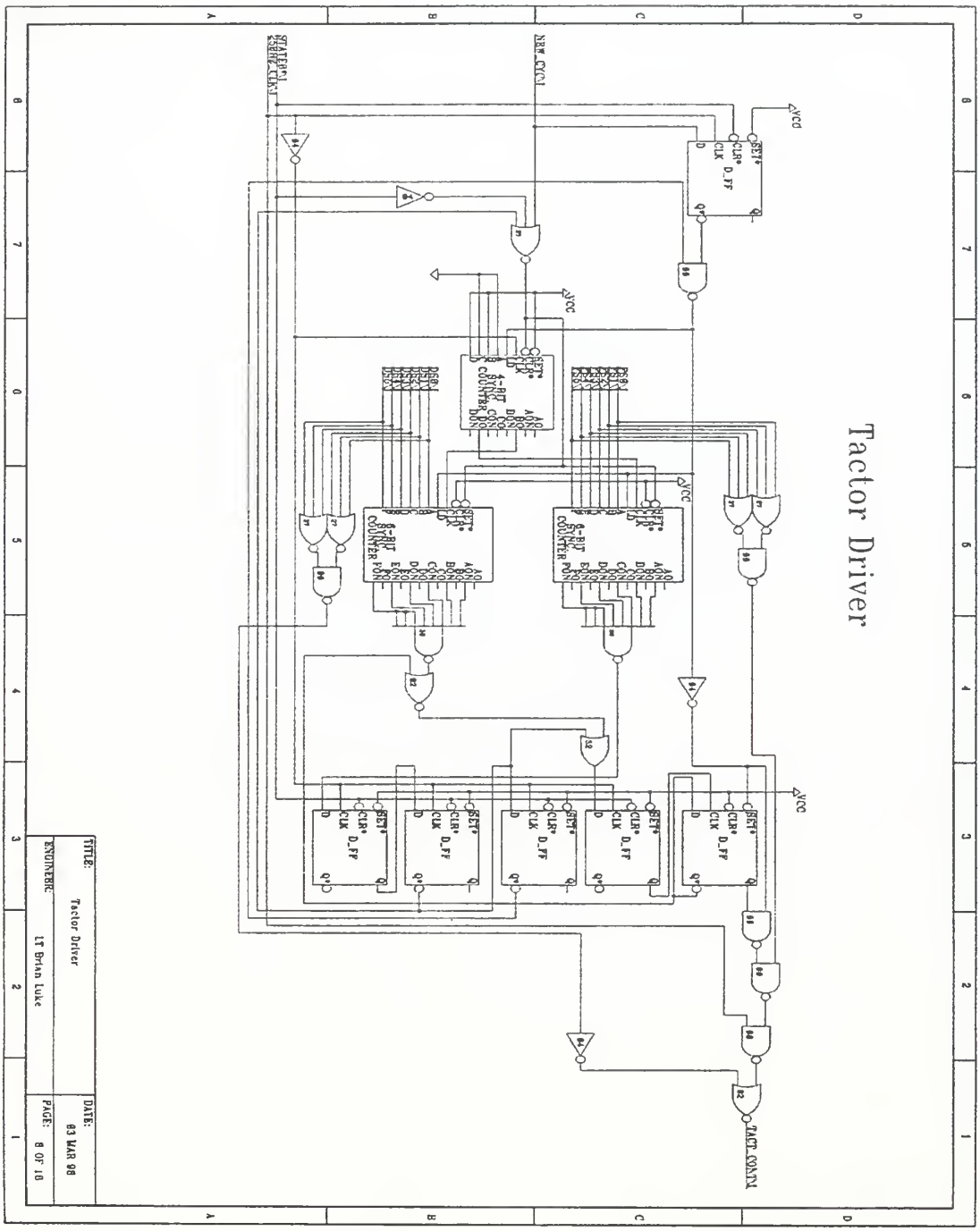



Figure 36. Tactor Driver

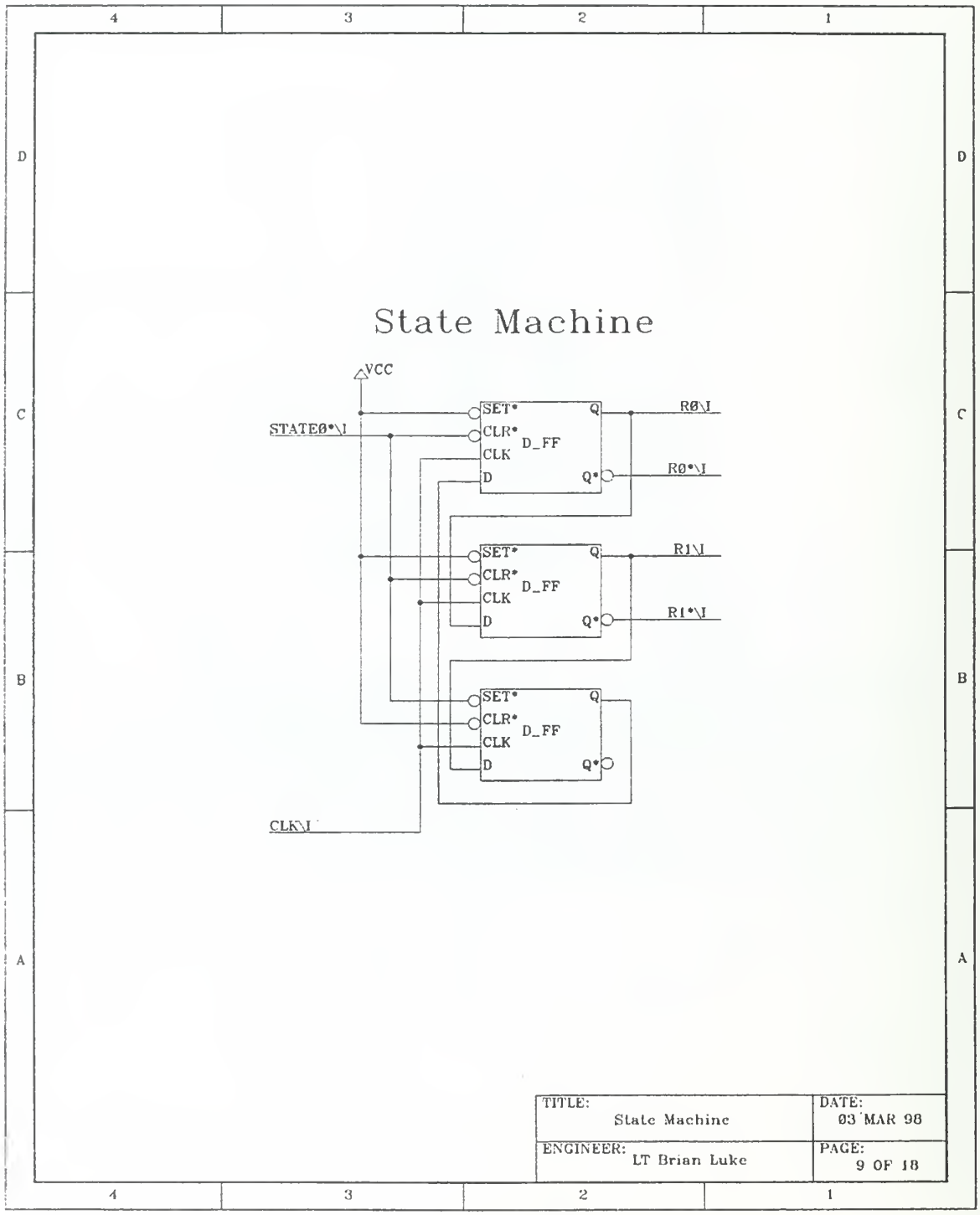



Figure 37. State Machine

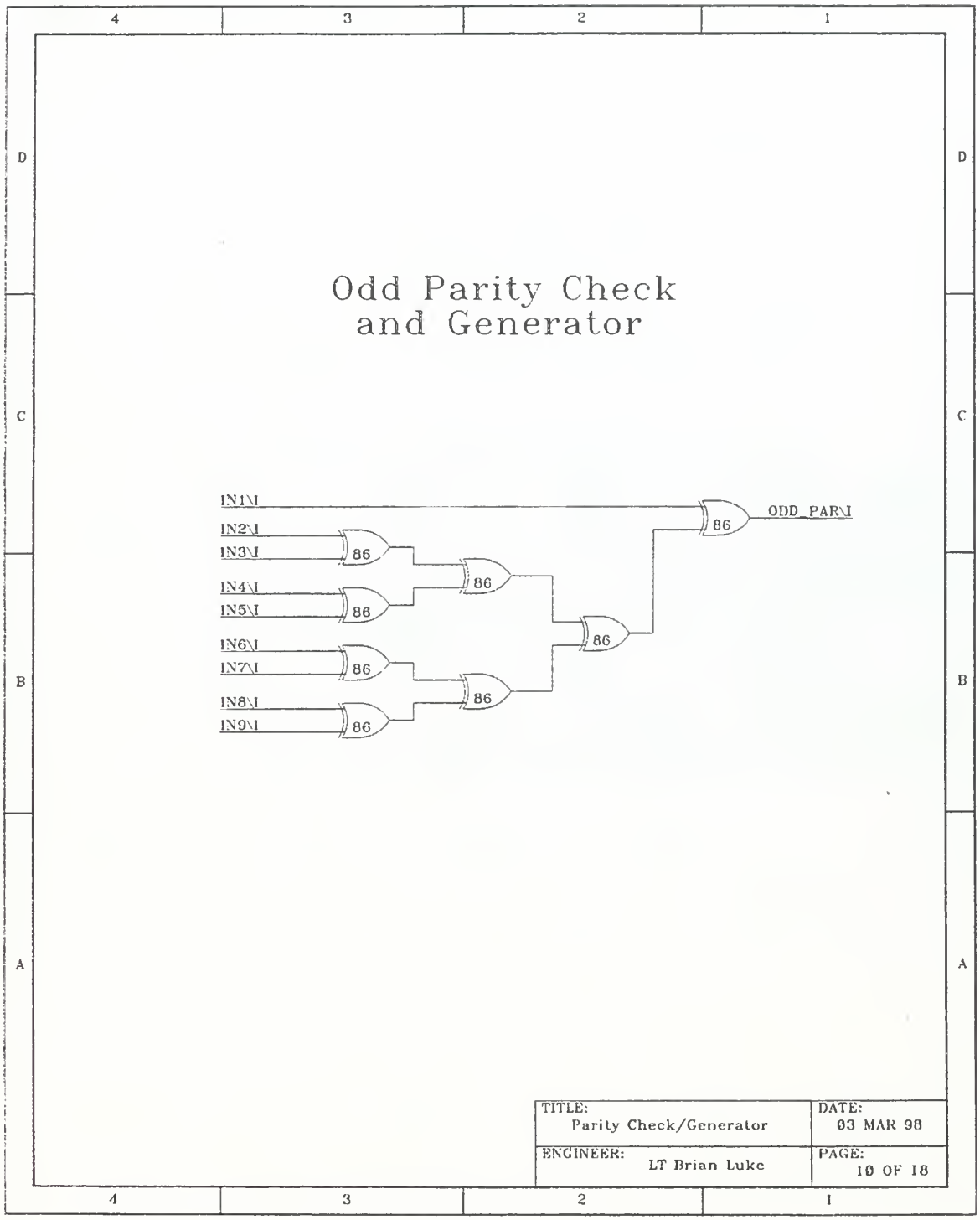



Figure 38. Odd Parity Check and Generator

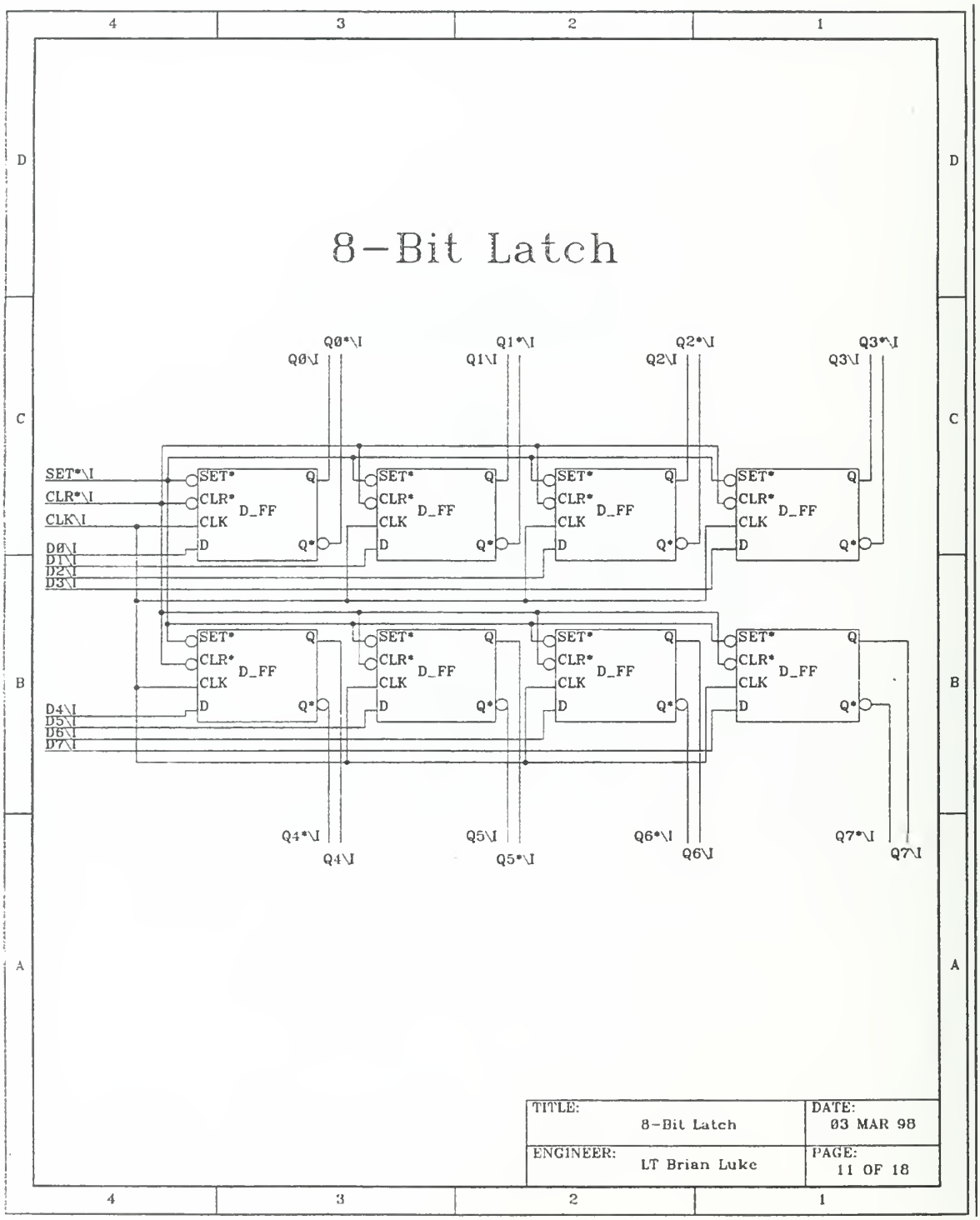



Figure 39. 8-Bit Latch

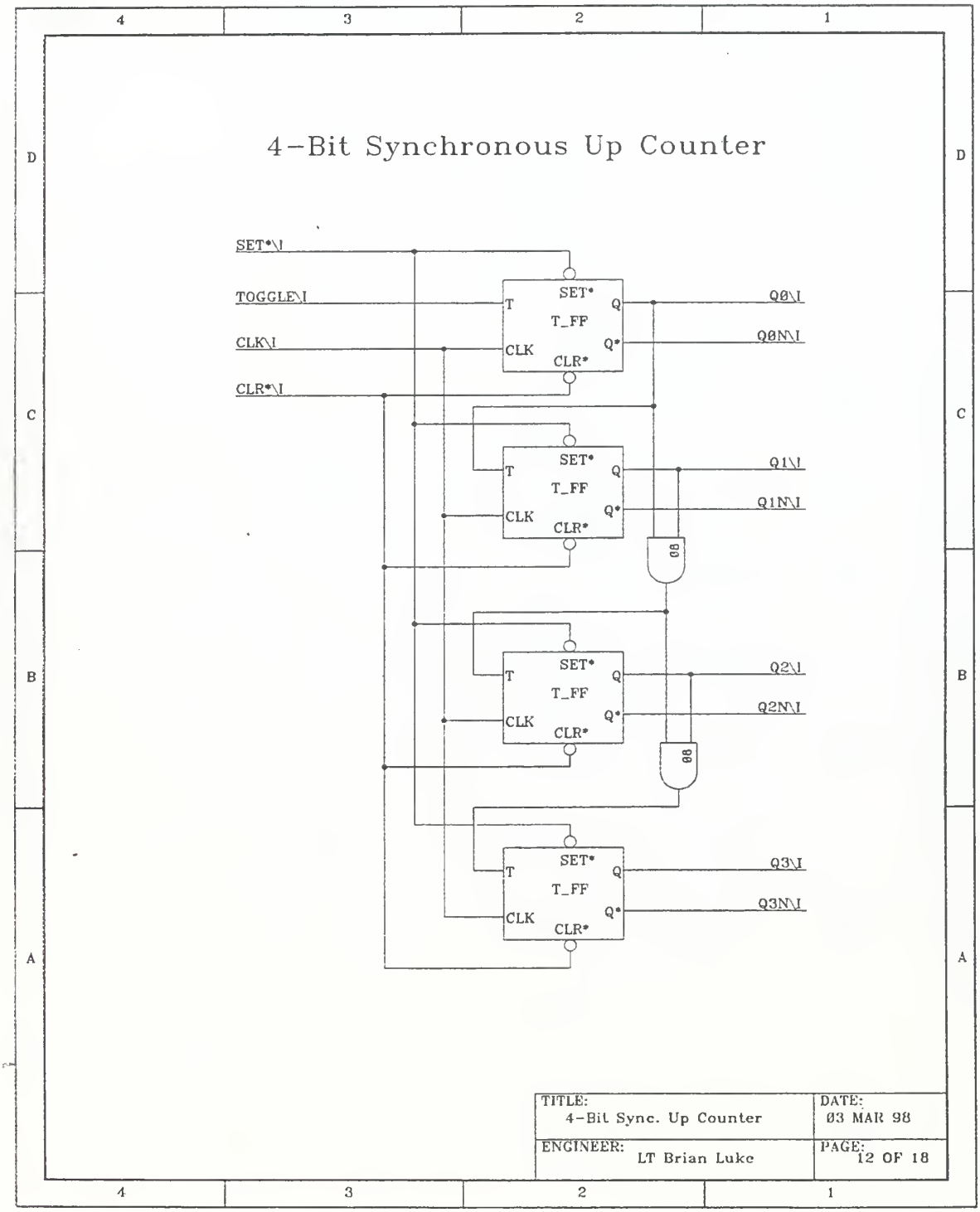



Figure 40. 4-Bit Synchronous Up Counter

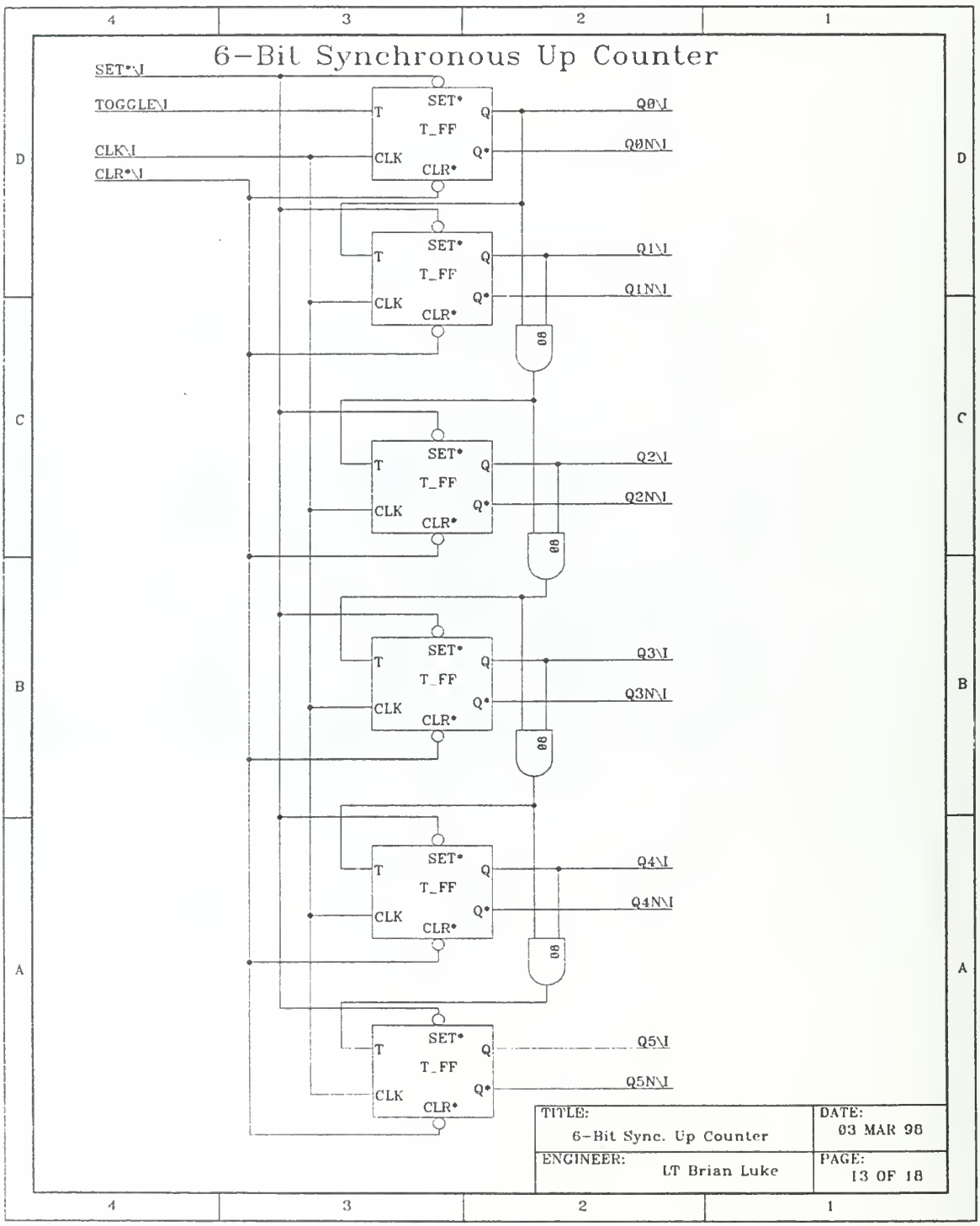



Figure 41. 6-Bit Synchronous Up Counter

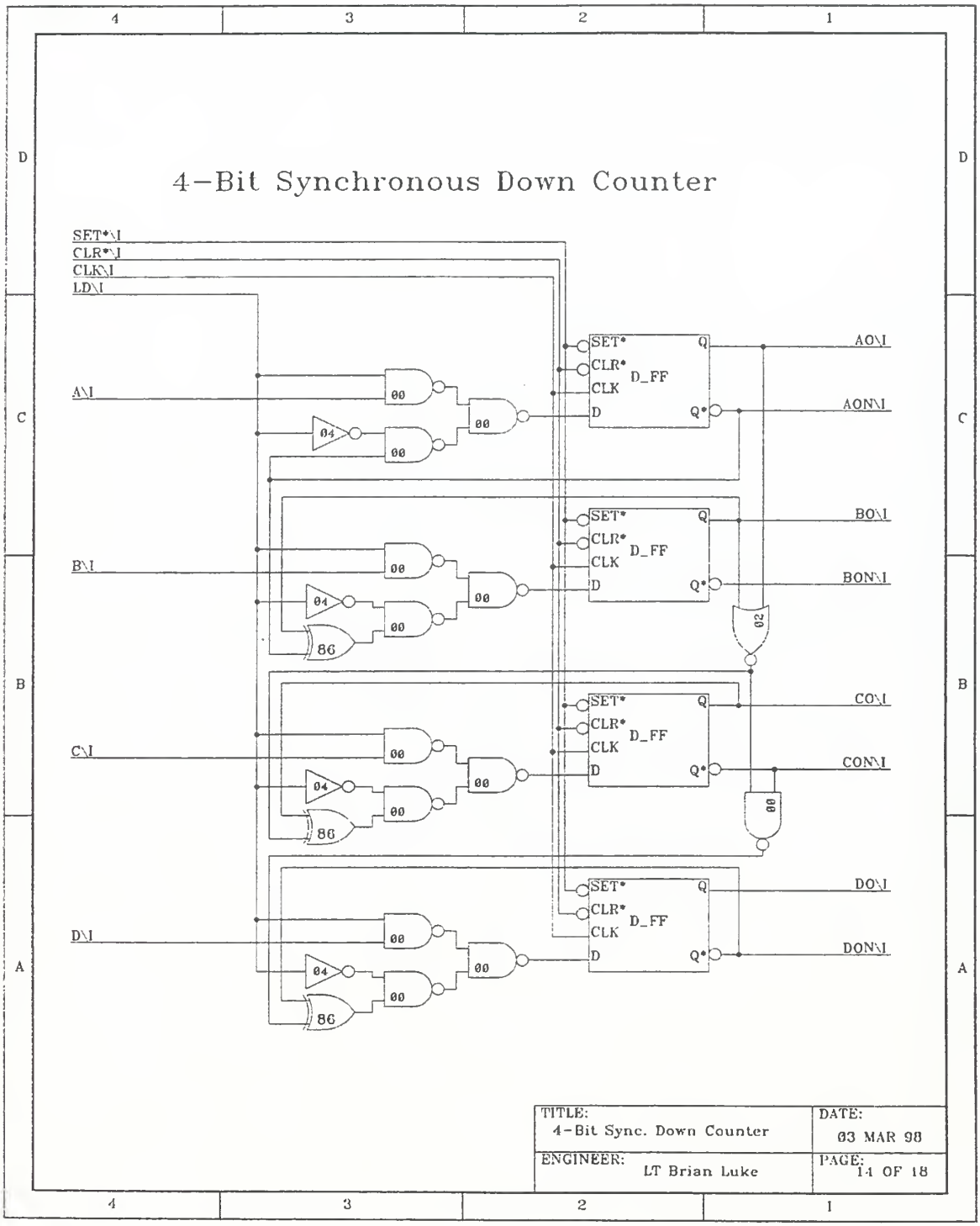



Figure 42. 4-Bit Synchronous Down Counter

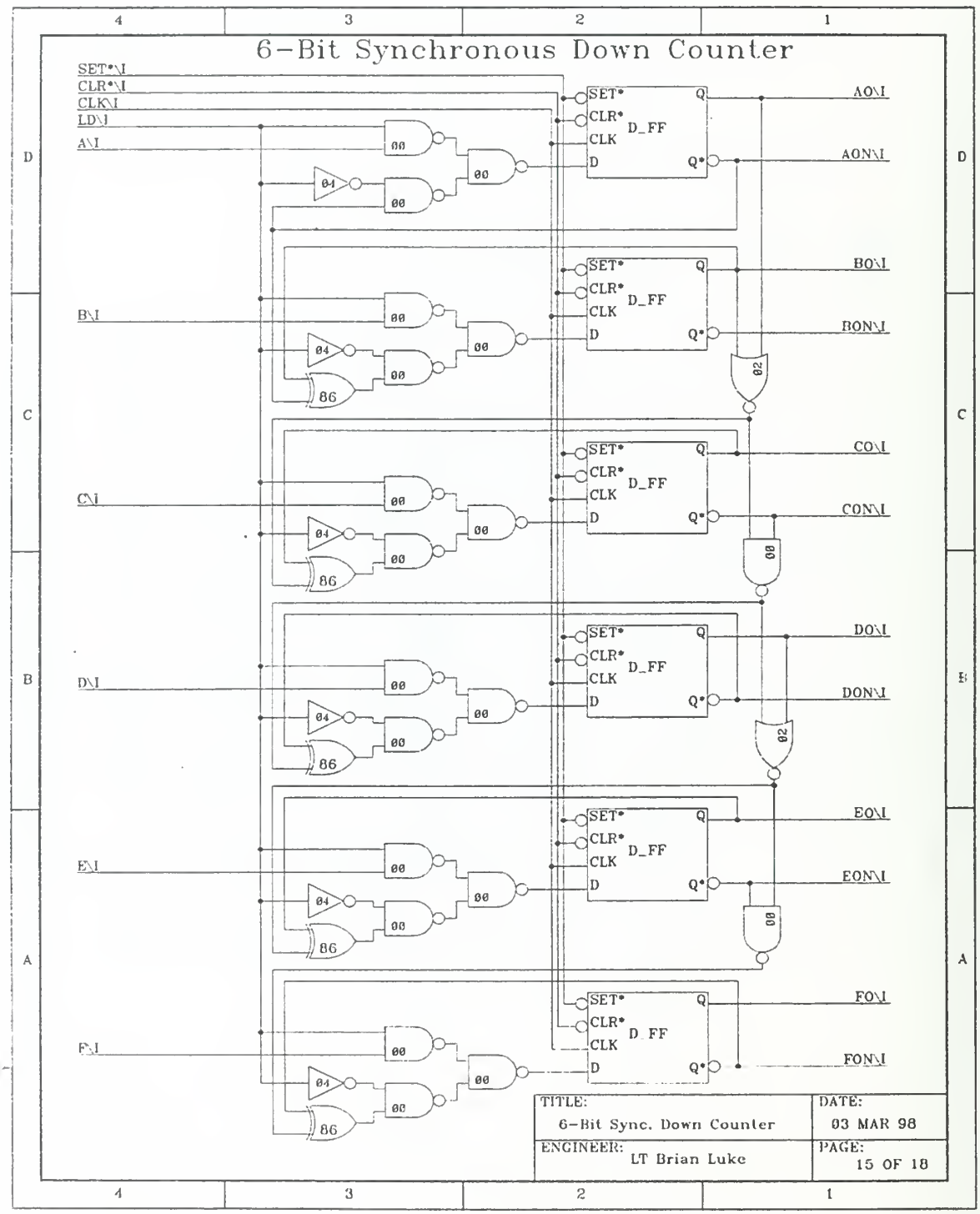



Figure 43. 6-Bit Synchronous Down Counter

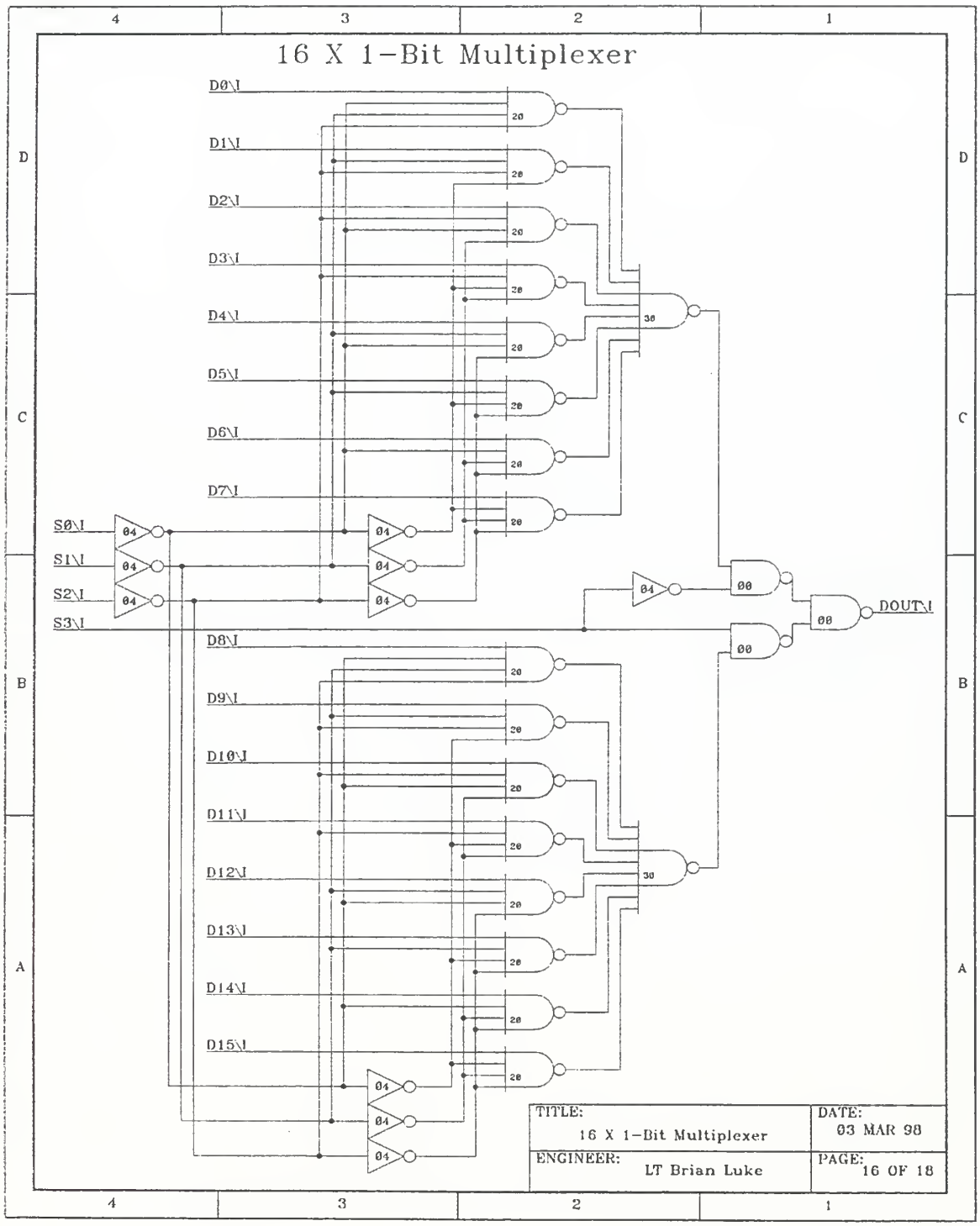



Figure 44. 16 x 1-Bit Multiplexer

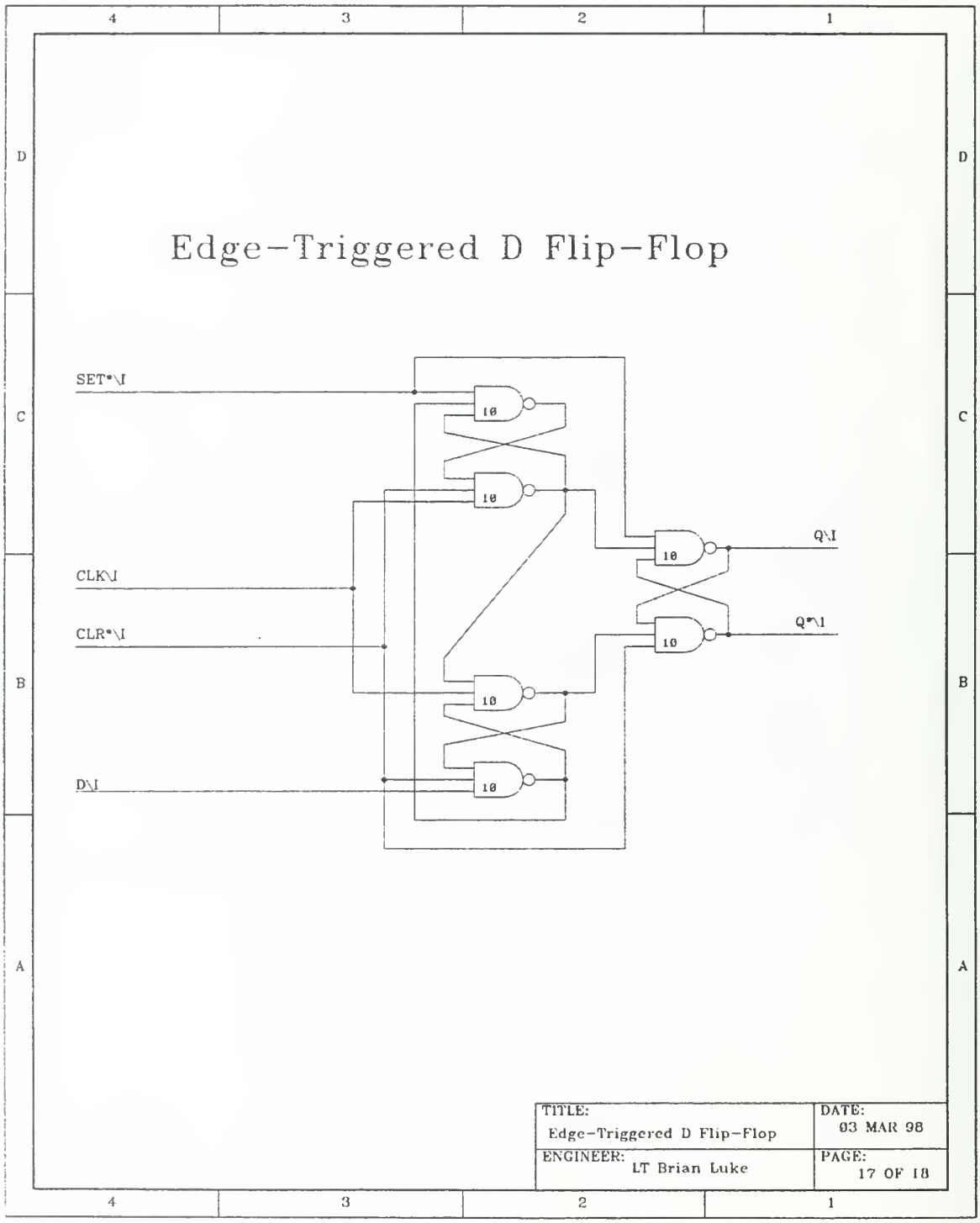



Figure 45. Edge-Triggered D-Flip-Flop

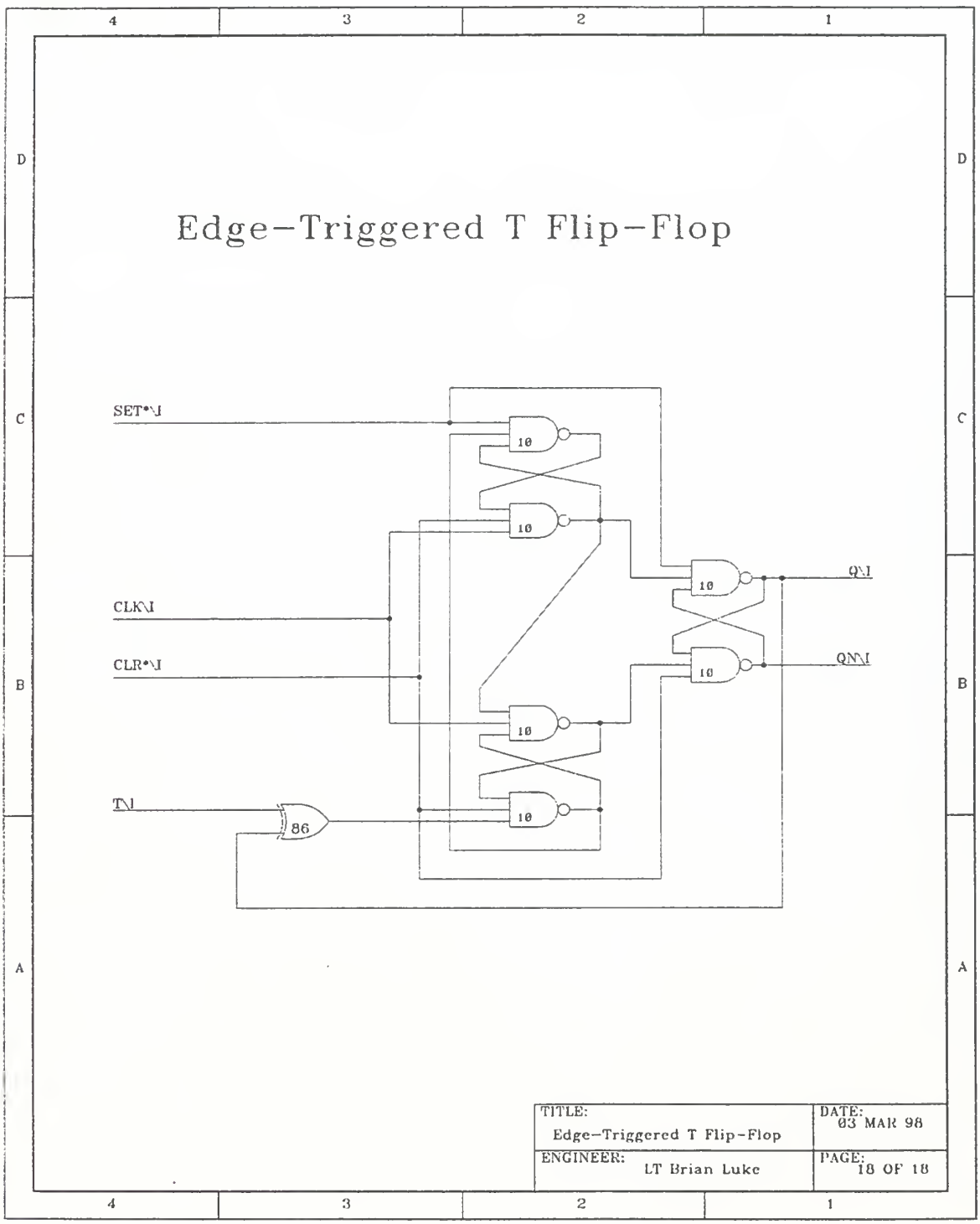



Figure 46. Edge-Triggered T-Flip-Flop

## APPENDIX B. MICROCONTROLLER SCHEMATICS

The diagrams in this appendix are the design schematics for the microcontroller design. The schematics are listed below.

1. MPC860 Connections
2. SRAM Memory Circuit
3. Flash Memory Circuit
4. EPROM Circuit
5. MIL-STD-1553 Interface
6. Serial Communications Circuit
7. Power Supply Circuit
8. Reset Circuit
9. Transceiver Circuit
10. Buffer Circuit
11. Clock Circuit
12. Pull-Up Resistors
13. 10-Pin Debug Connector

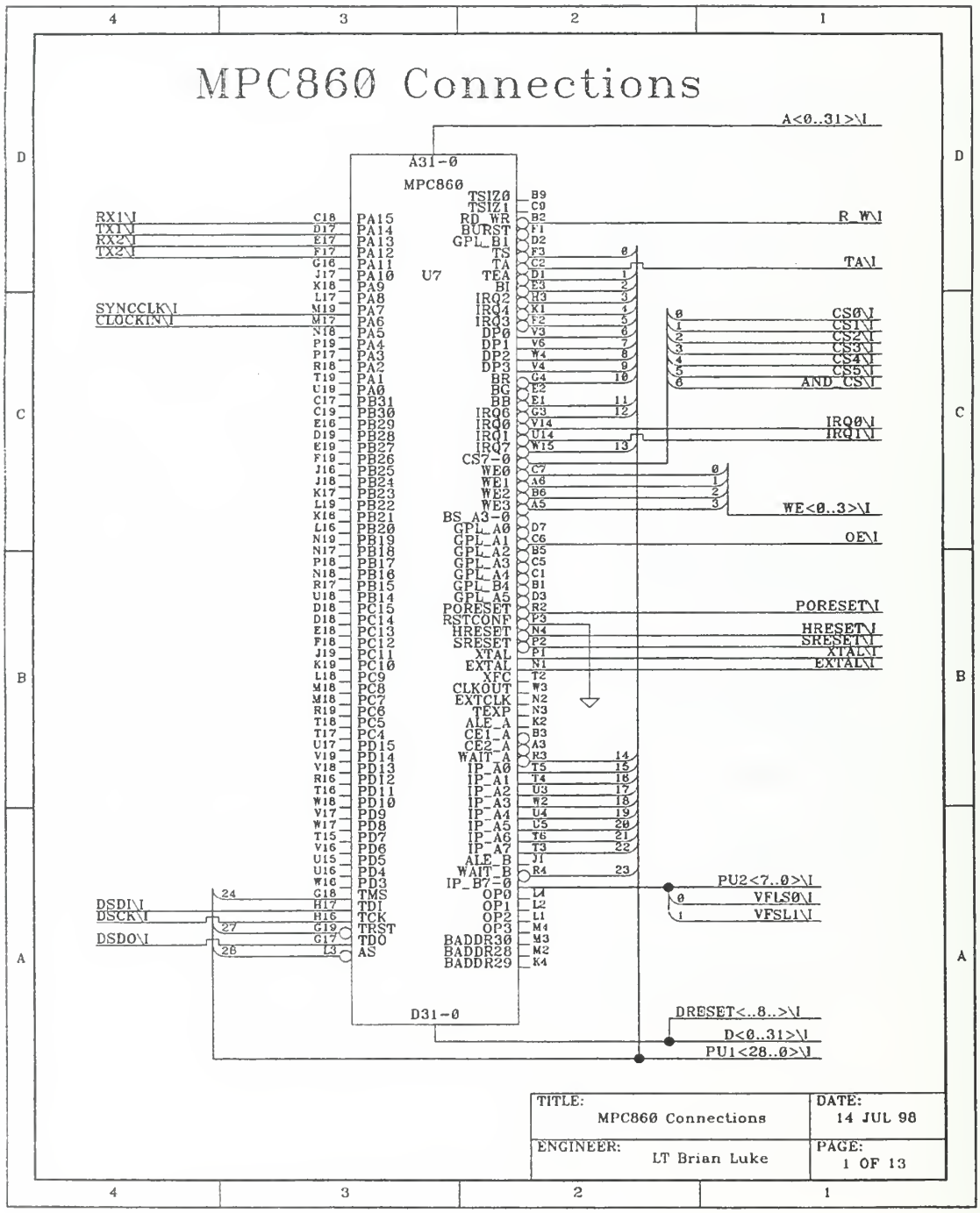



Figure 47. MPC860 Connections

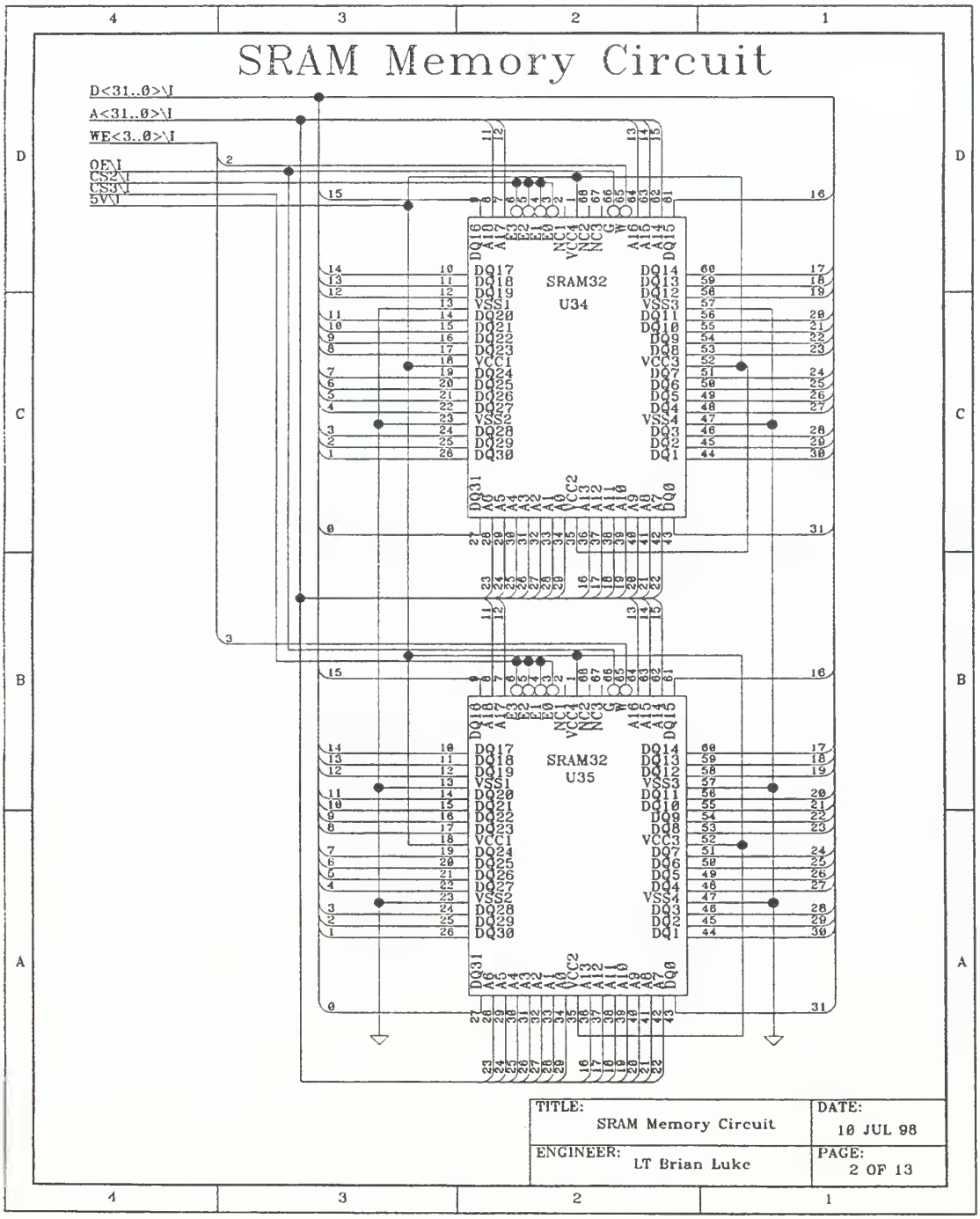



Figure 48. SRAM Memory Circuit

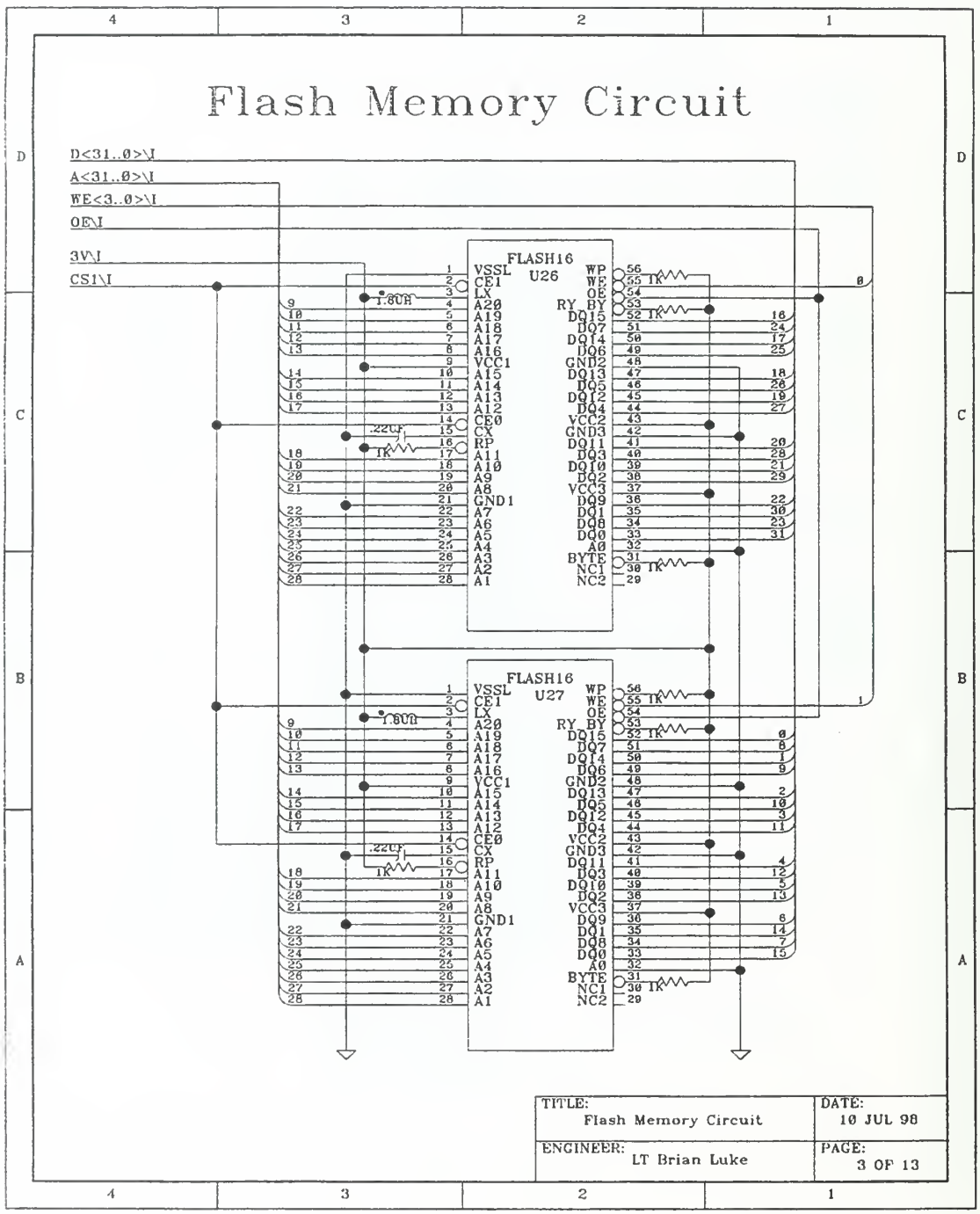



Figure 49. Flash Memory Circuit

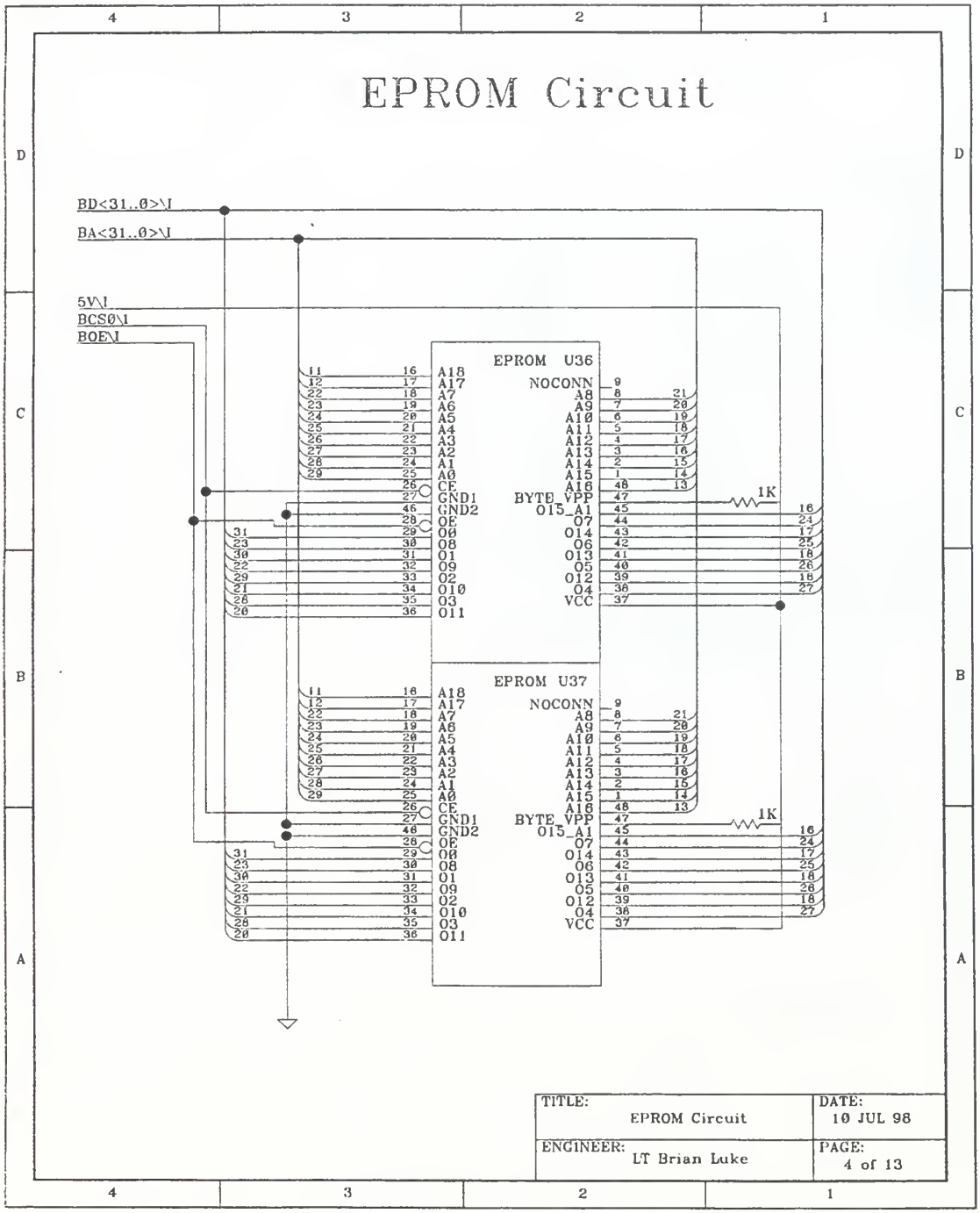



Figure 50. EPROM Memory Circuit

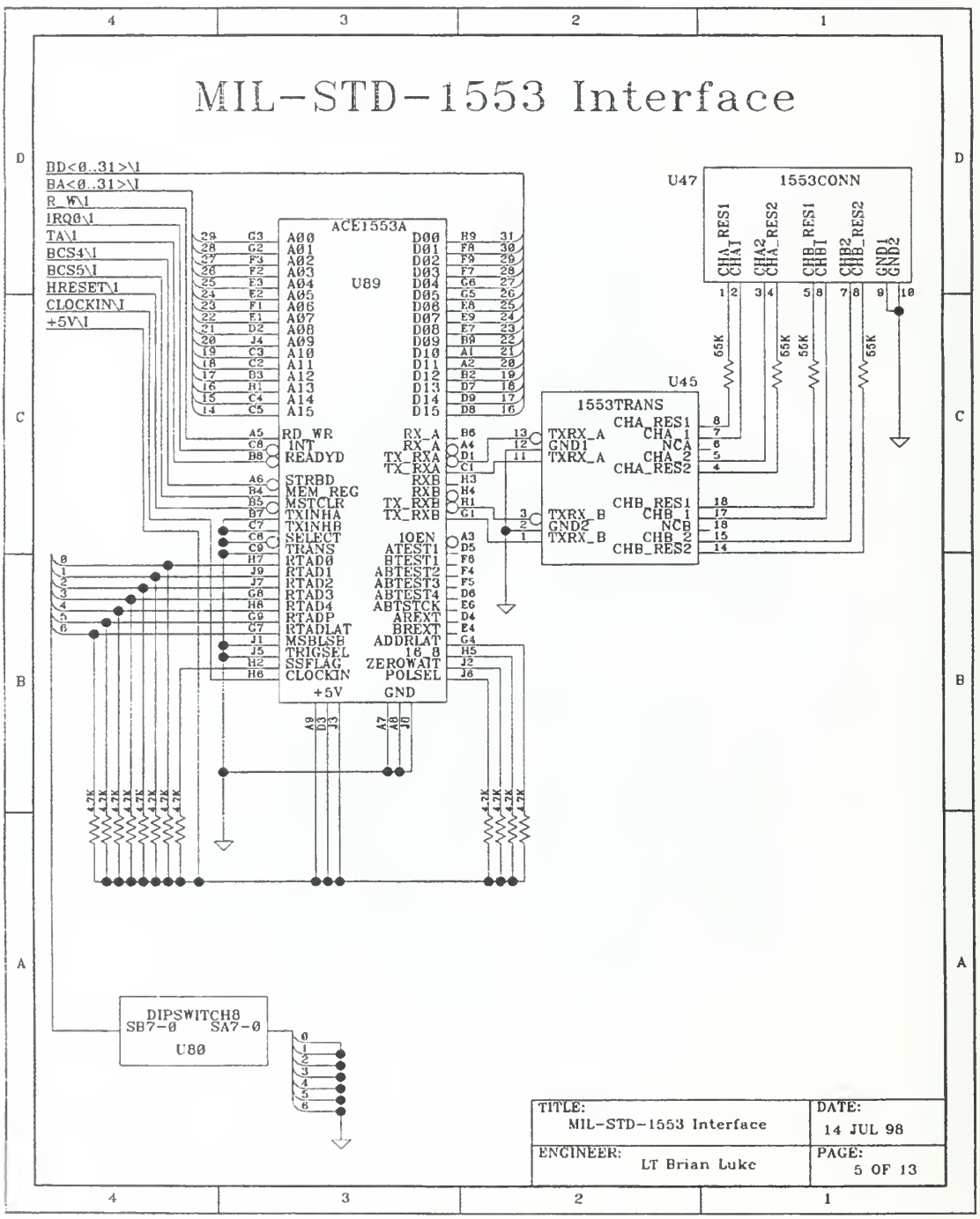



Figure 51. MIL-STD-1553 Interface

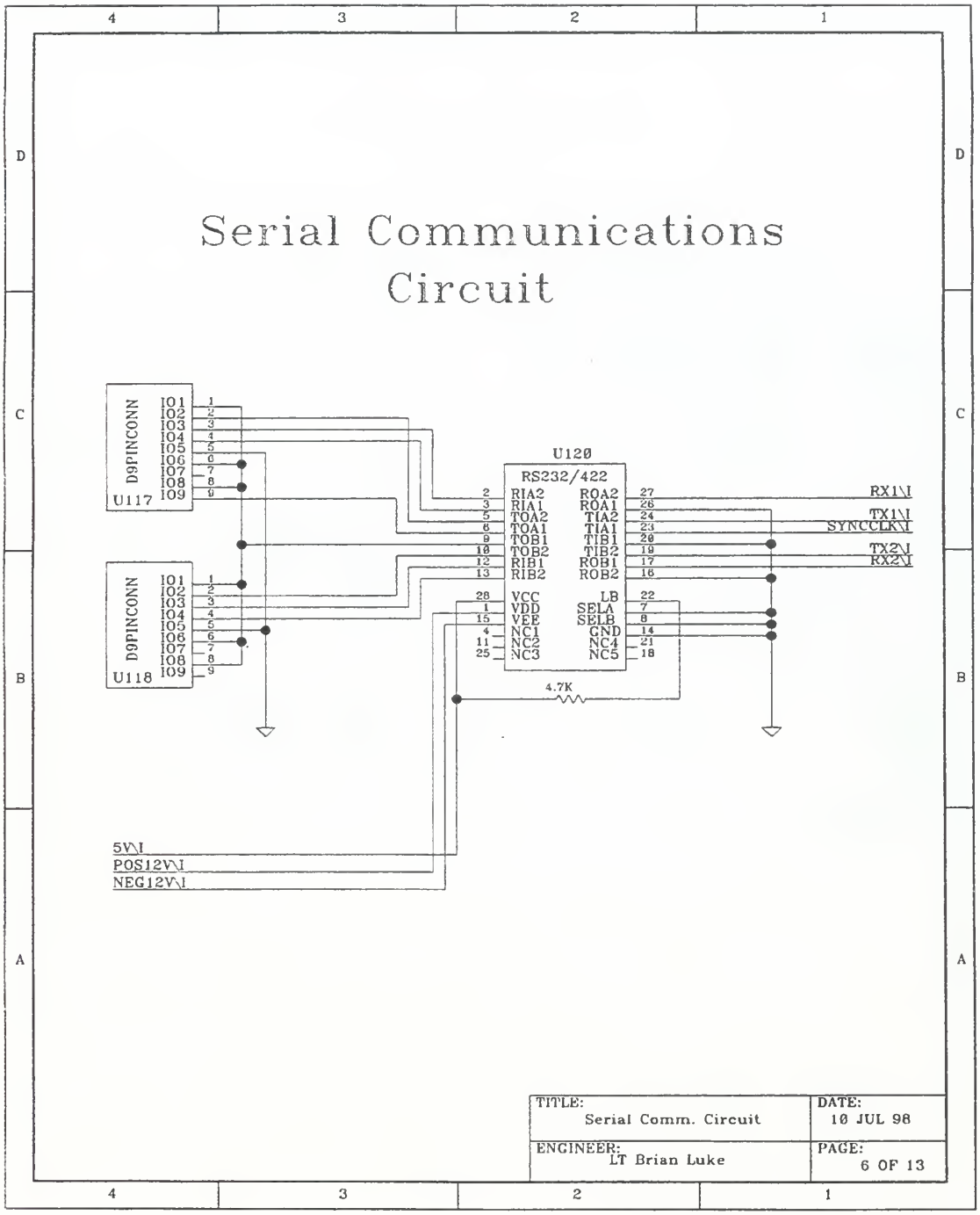



Figure 52. Serial Communications Circuit

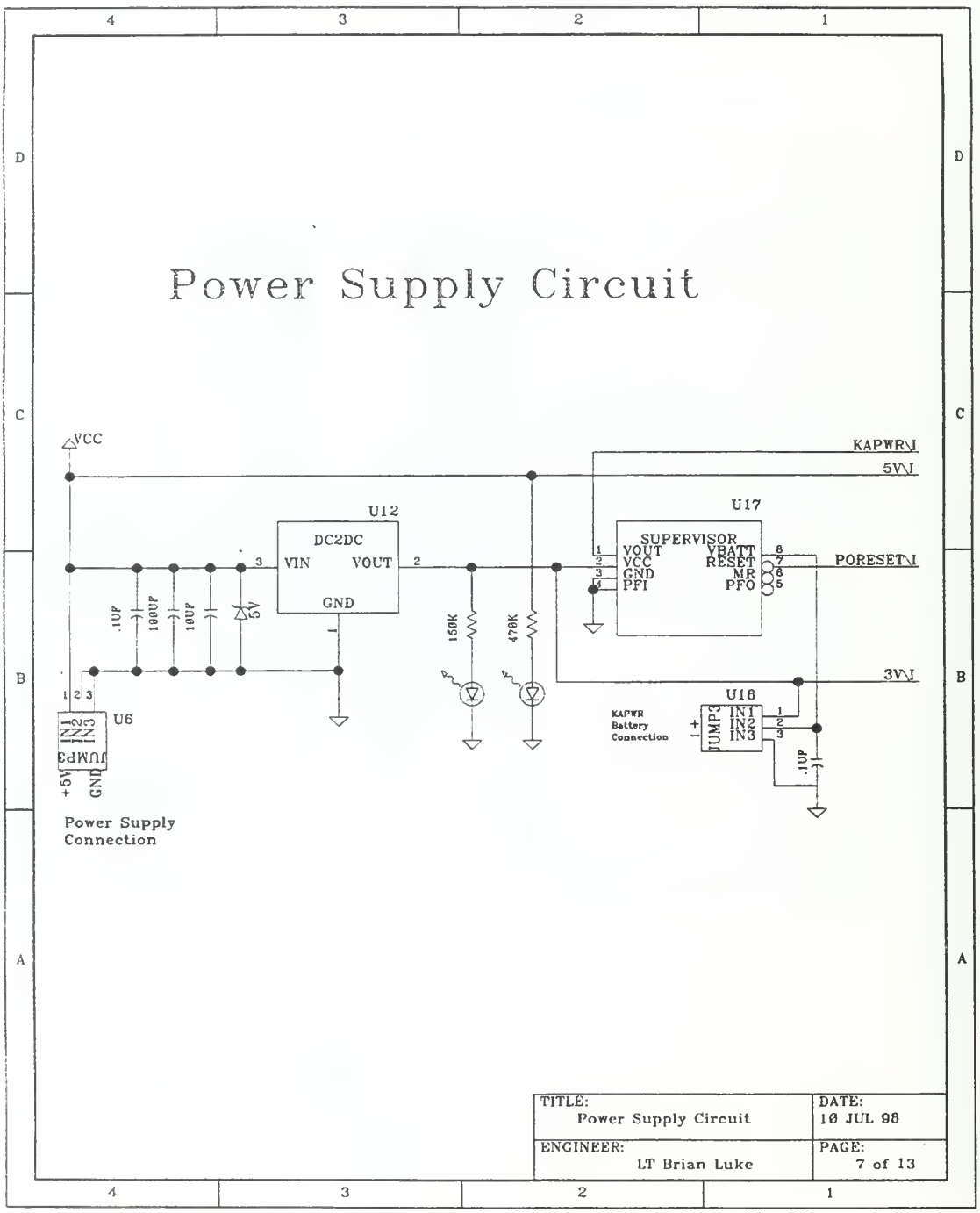



Figure 53. Power Supply Circuit

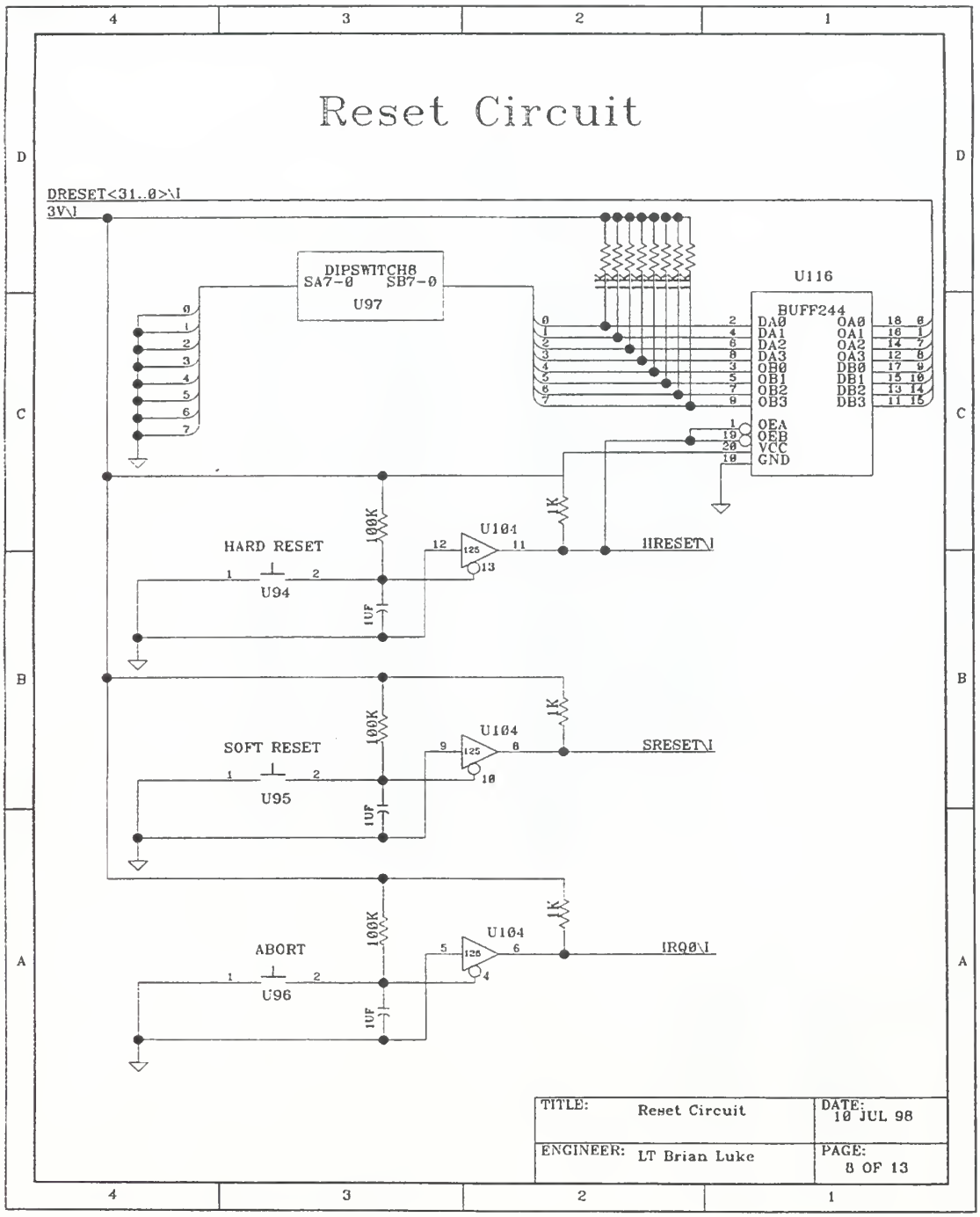



Figure 54. Reset Circuit

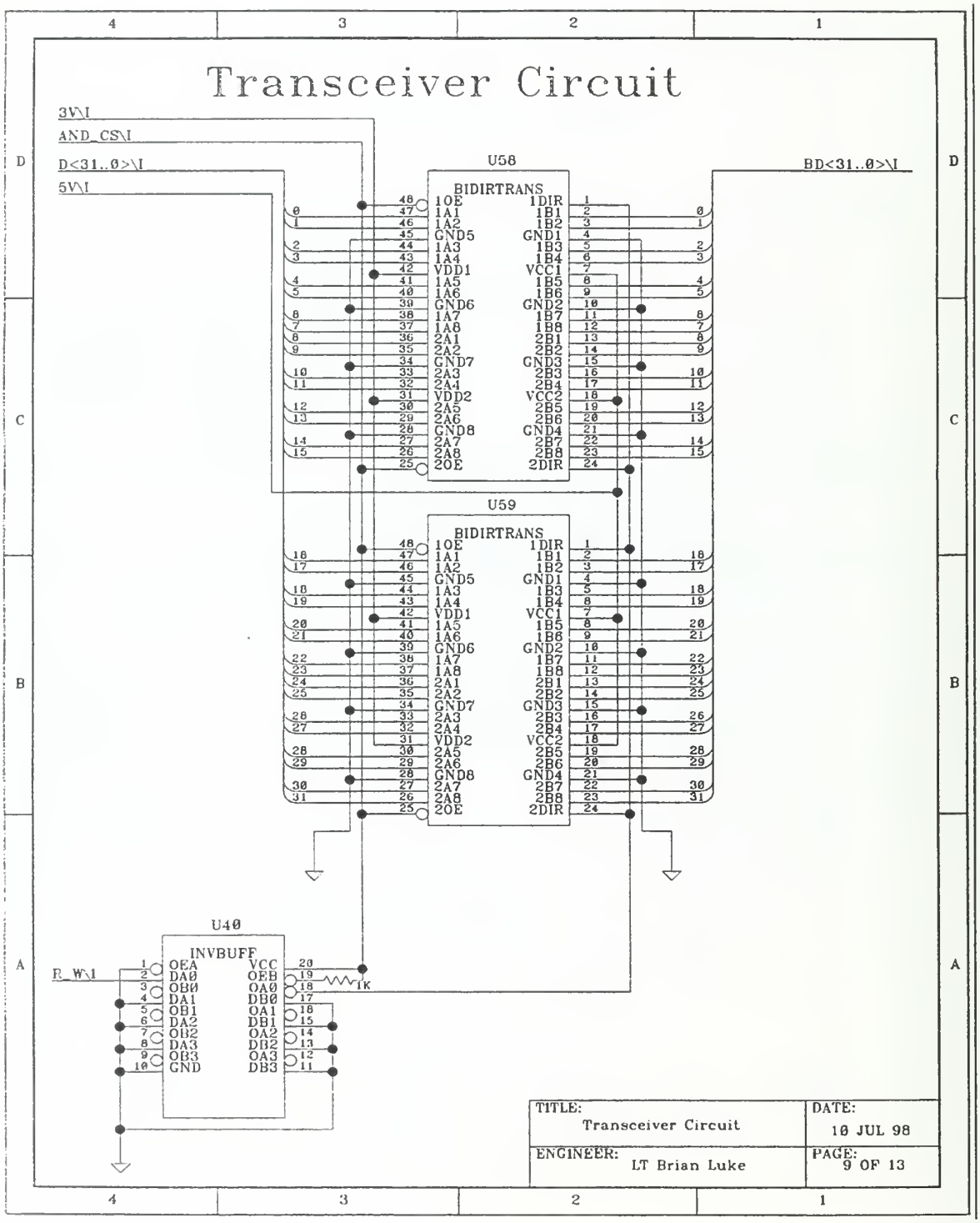



Figure 55. Transceiver Circuit

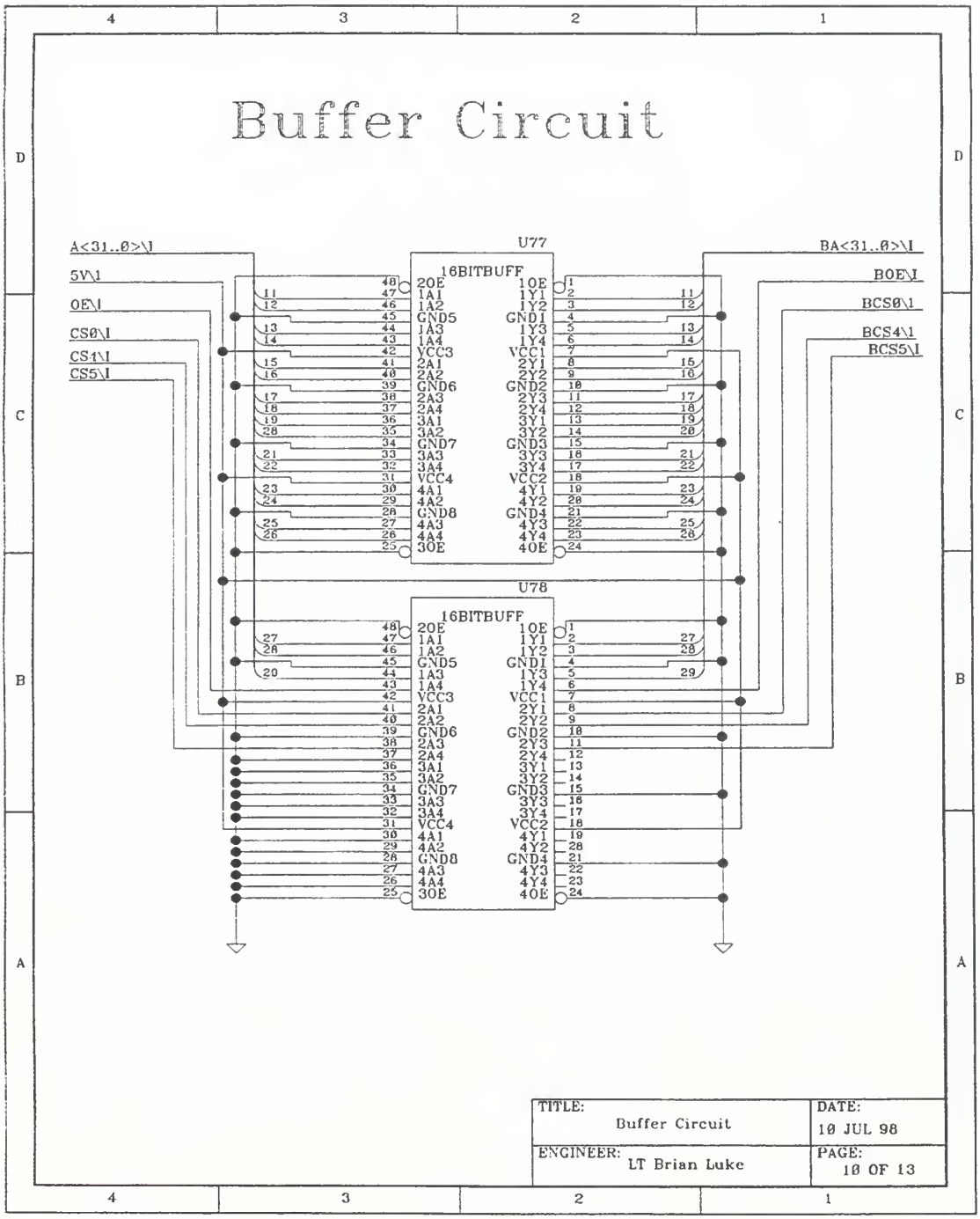



Figure 56. Buffer Circuit

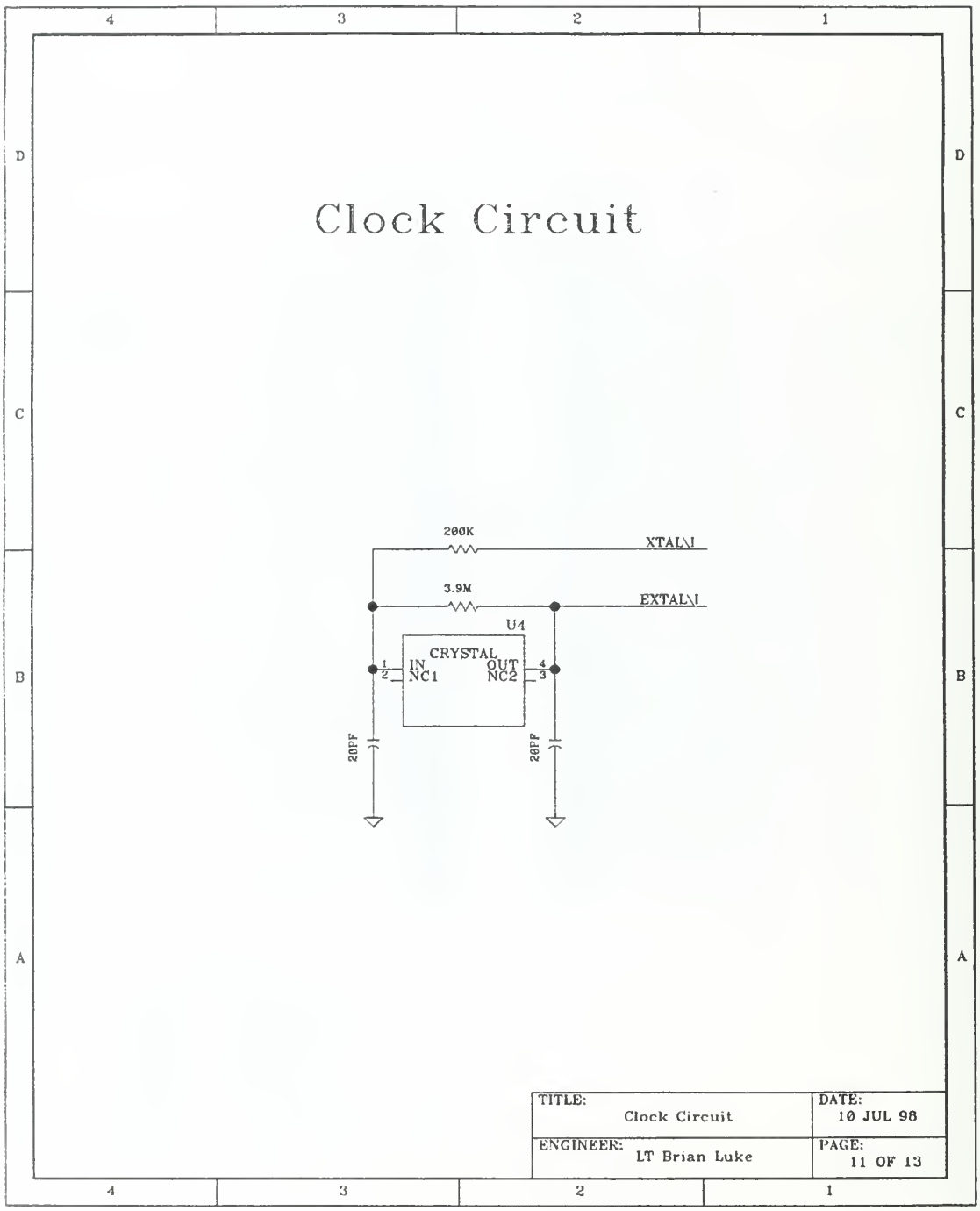



Figure 57. Clock Circuit

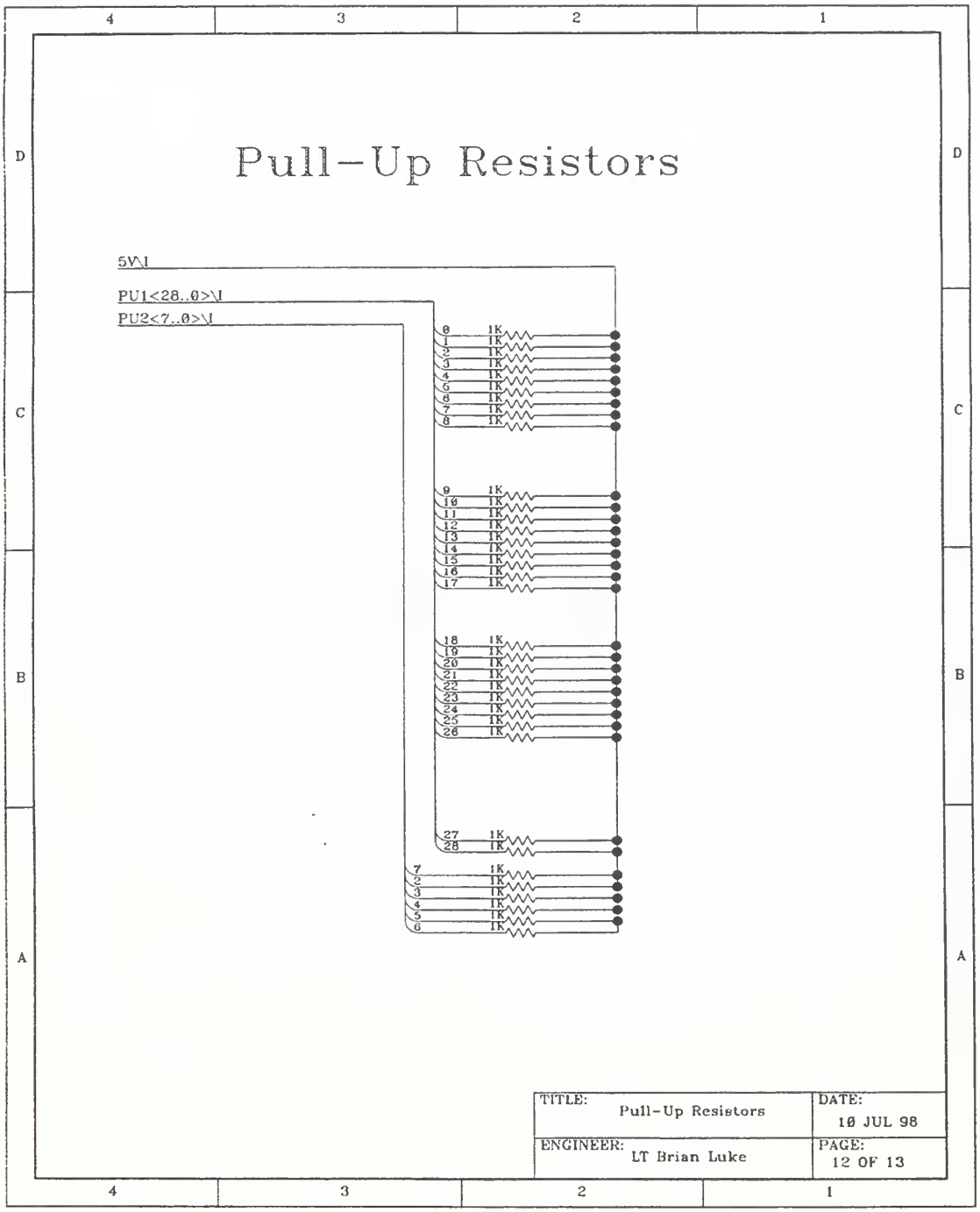



Figure 58. Pull-Up Resistors

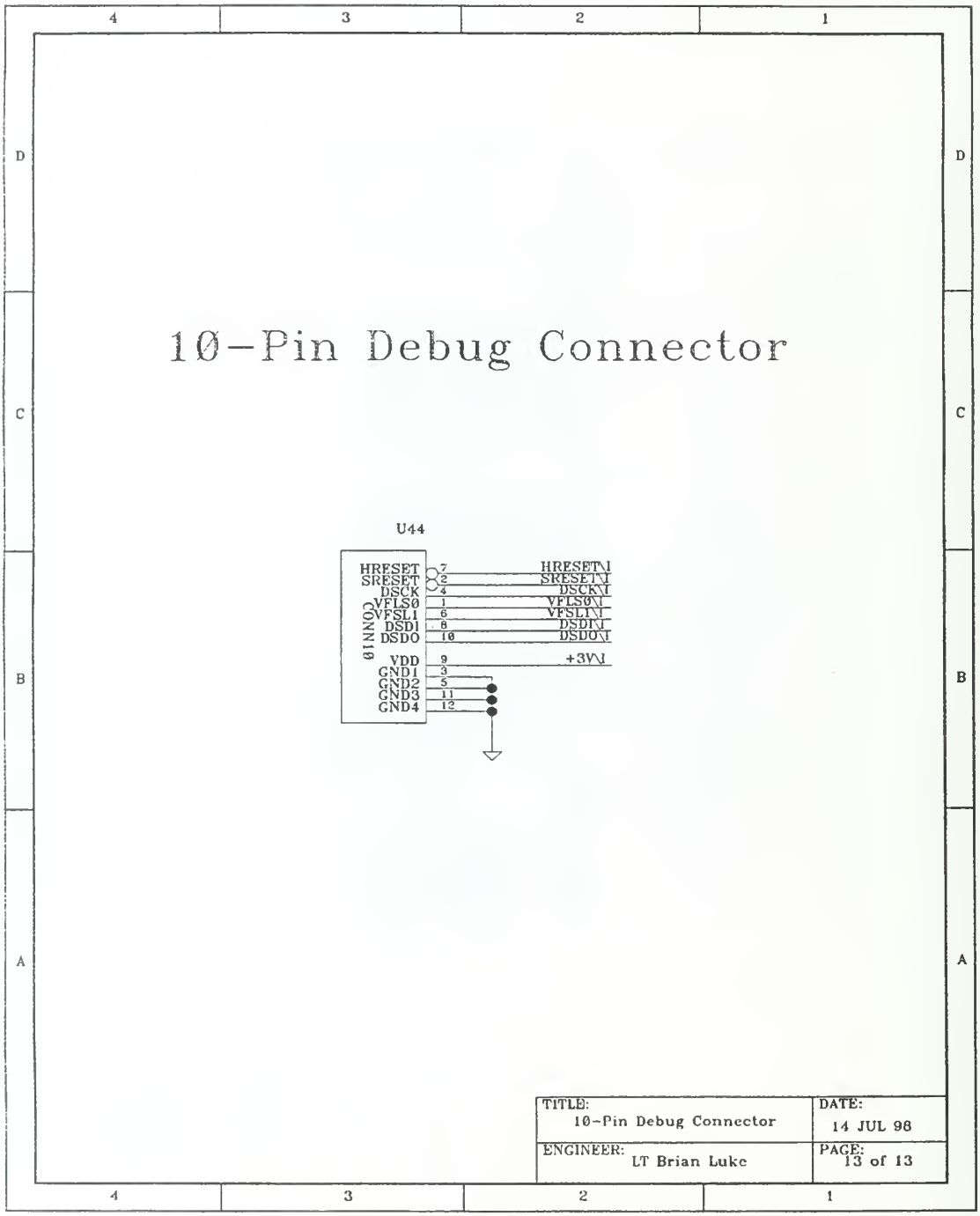



Figure 59. 10-Pin Debug Connector

## APPENDIX C. MICROCONTROLLER PRINTED CIRCUIT BOARD LAYOUT

The diagrams in this appendix are the components of the printed circuit board (PCB) design for the microcontroller system. The diagrams listed below are scale representations of the PCB layout.

1. PCB design with all parts and etch
2. PCB design with parts only
3. MPC860 with pull-up resistors and debug connector
4. Memory components and connections
5. MIL-STD-1553 components and connections
6. Serial Communications and clock components
7. Power Supply components
8. Reset circuit components
9. EPROM, buffer, and transceiver components

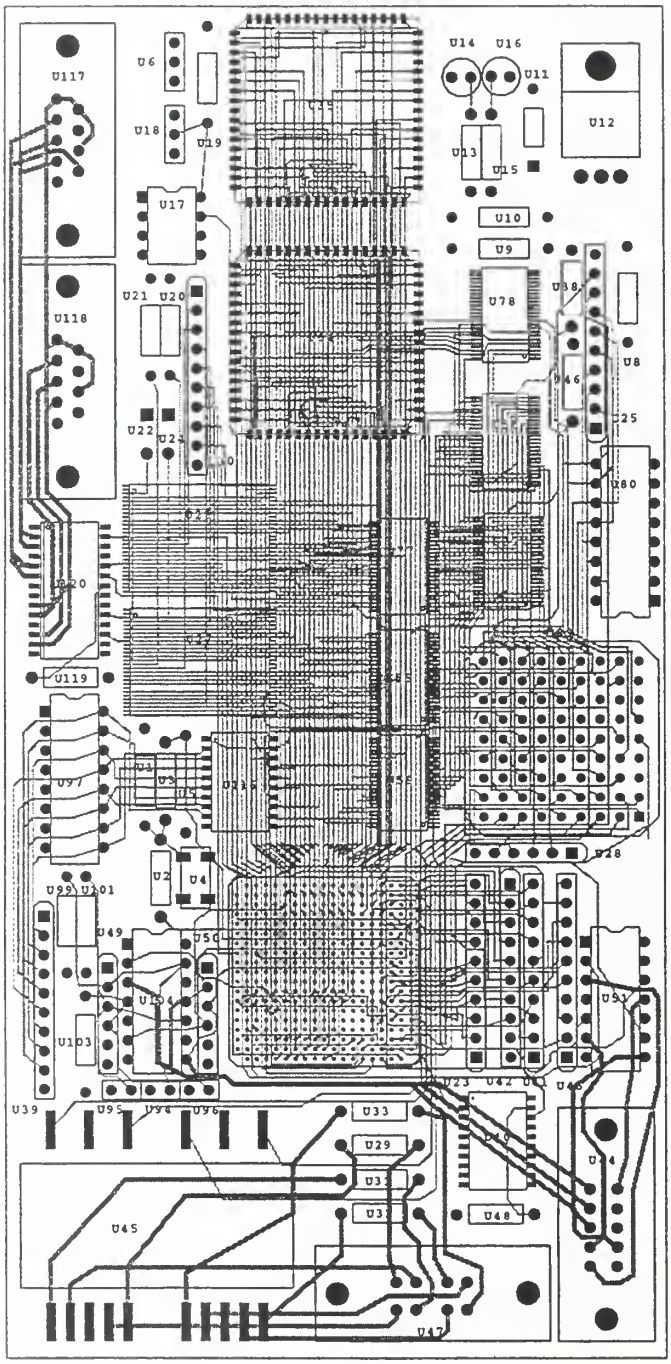



Figure 60. PCB design with all parts and etch

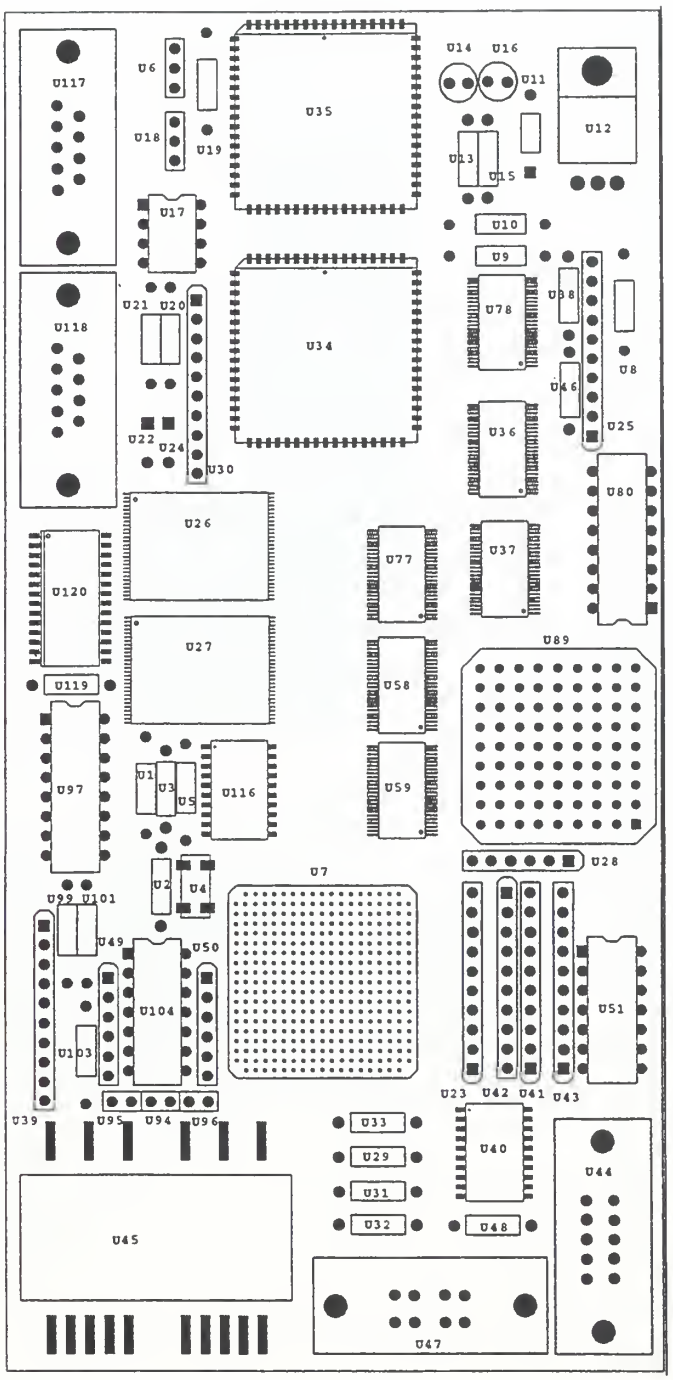



Figure 61. PCB design with parts only

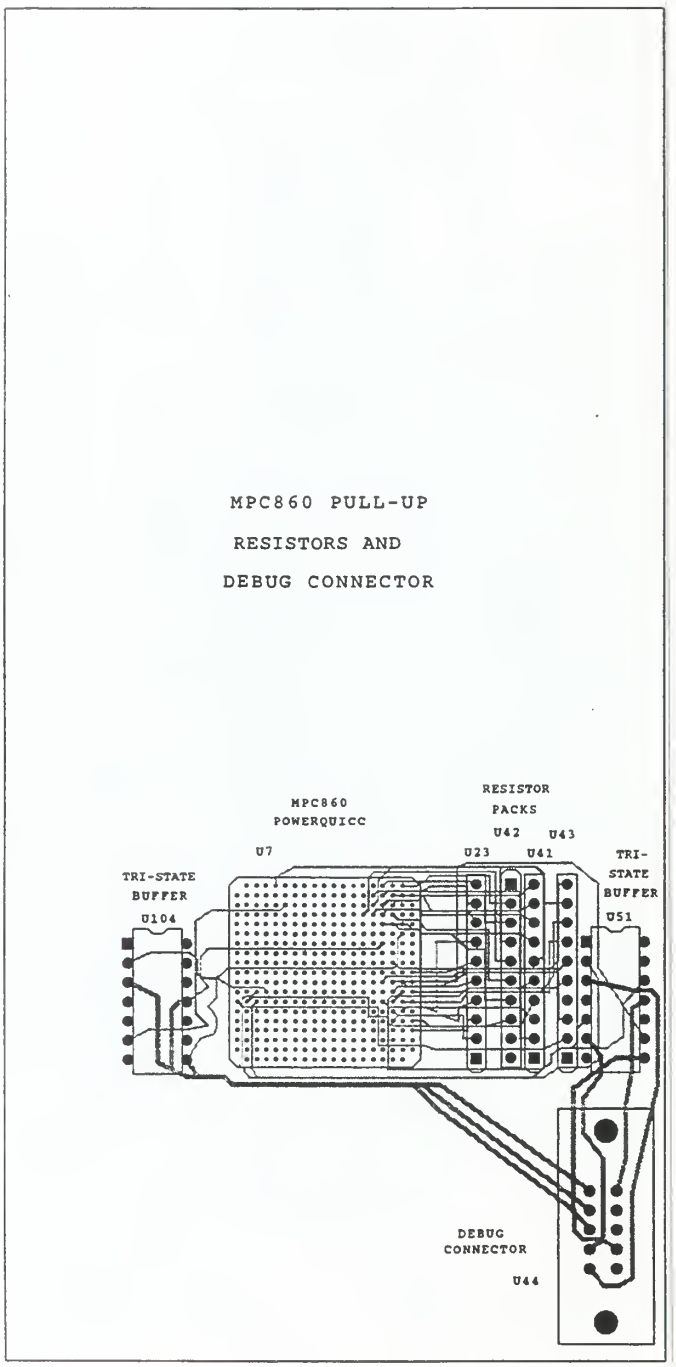



Figure 62. MPC860, pull-up resistors, and debug connector

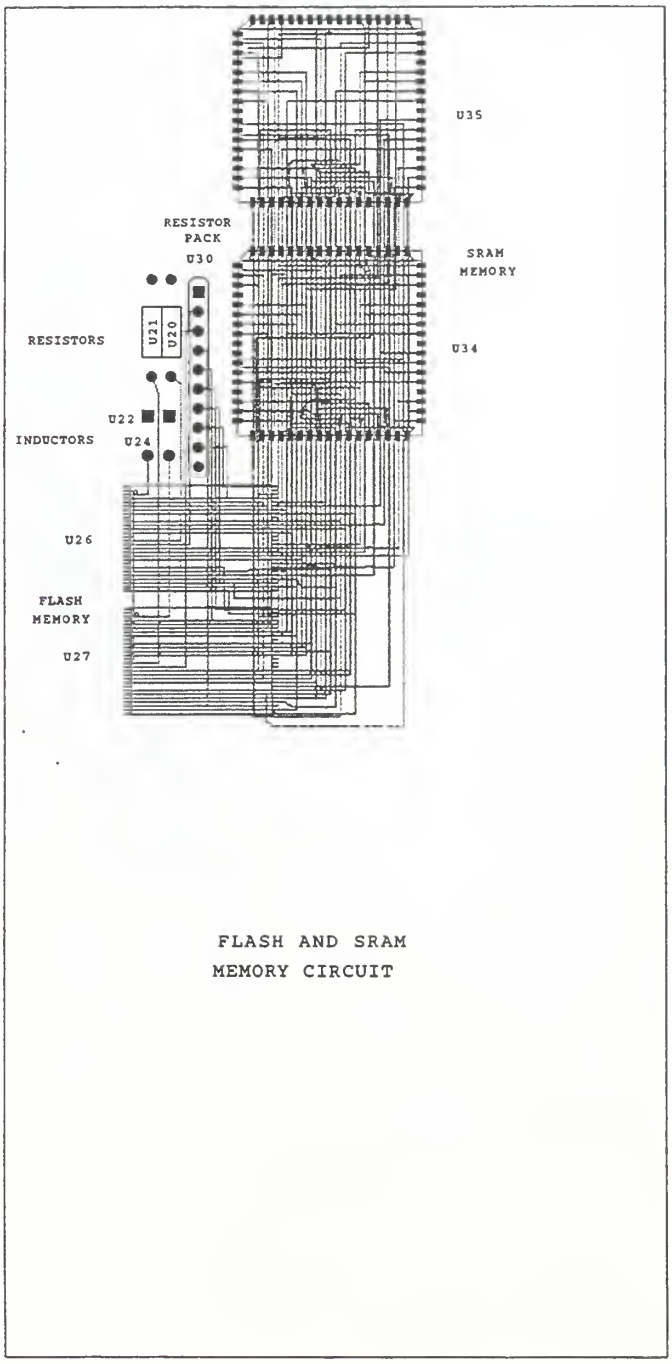



Figure 63. SRAM and Flash memory components

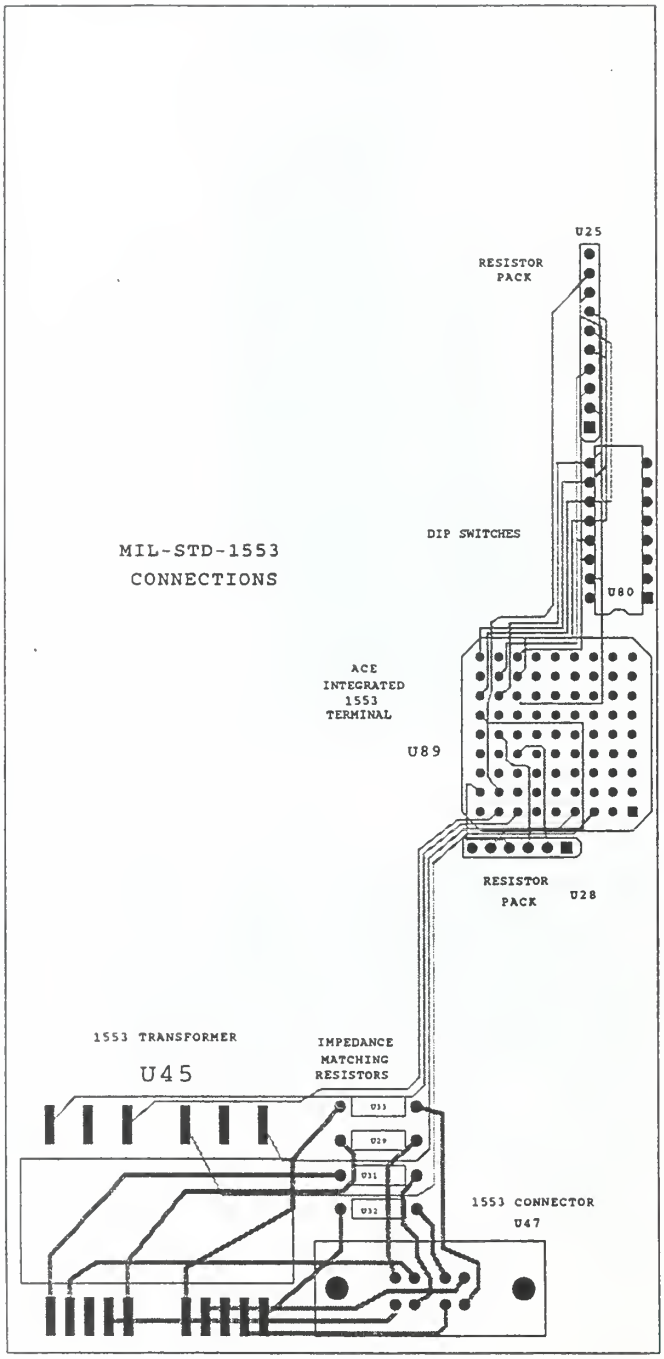



Figure 64. MIL-STD-1553 components

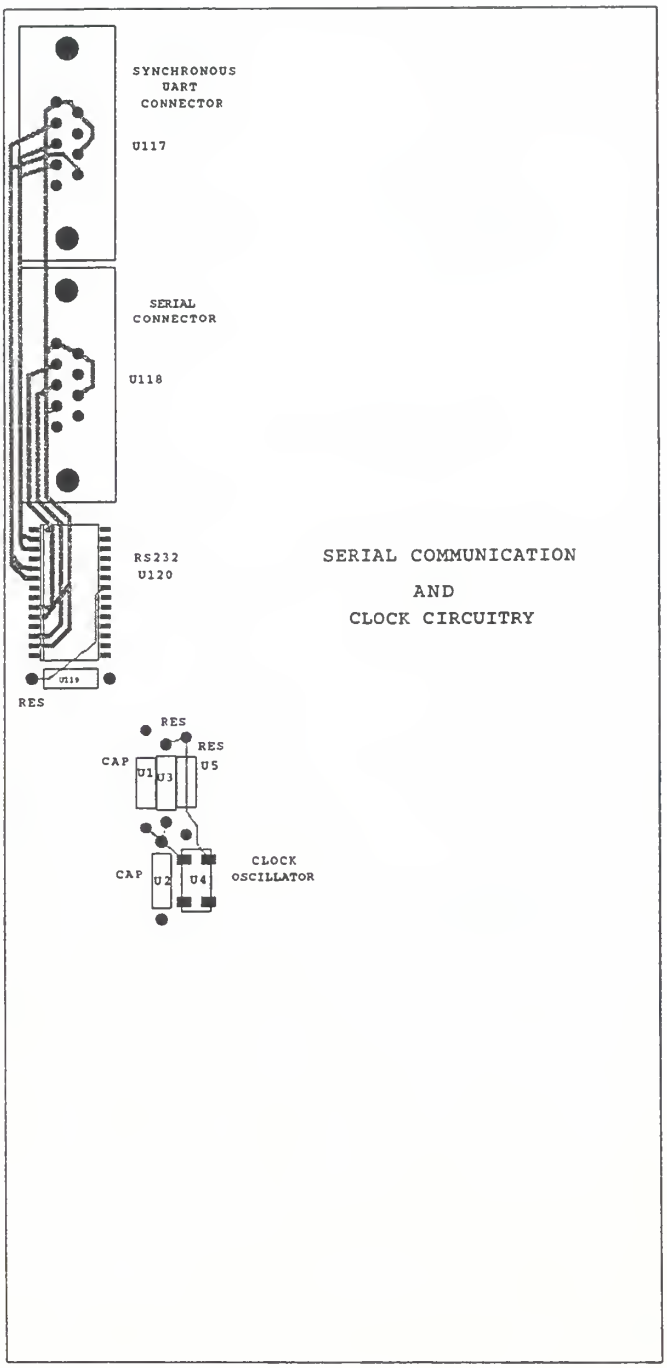



Figure 65. Serial communications and clock components

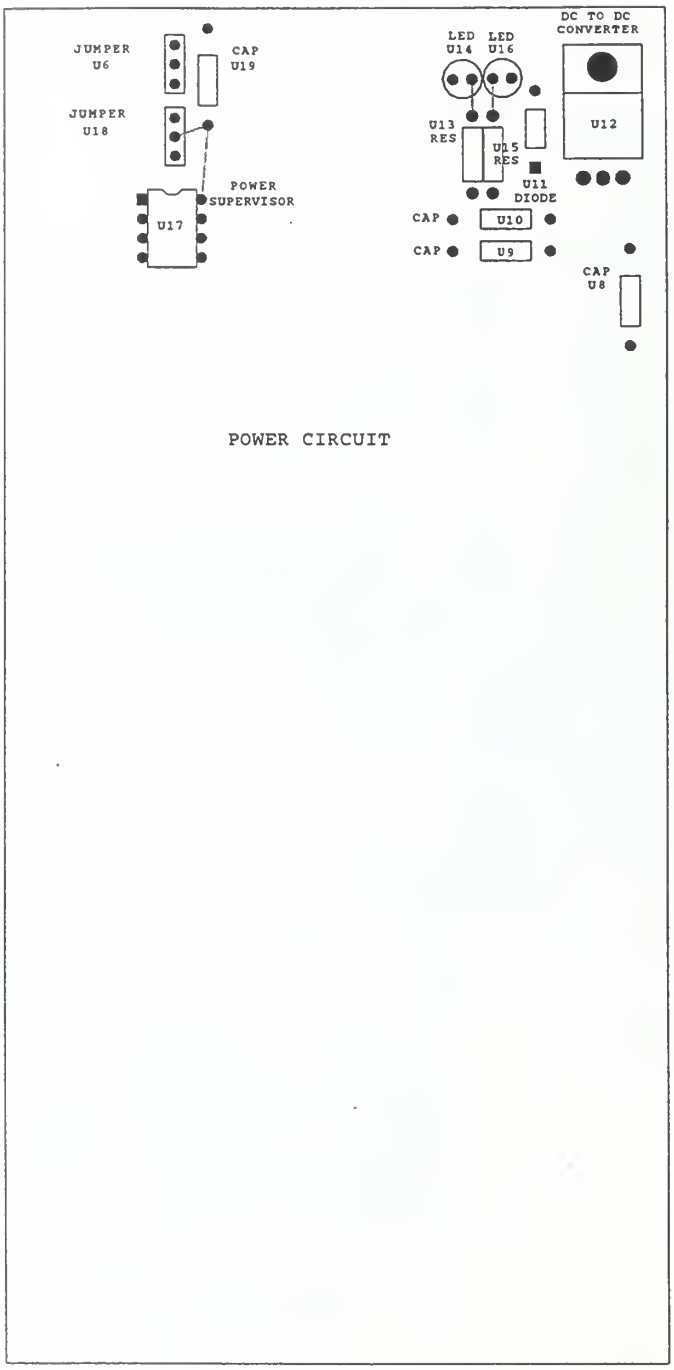



Figure 66. Power supply components

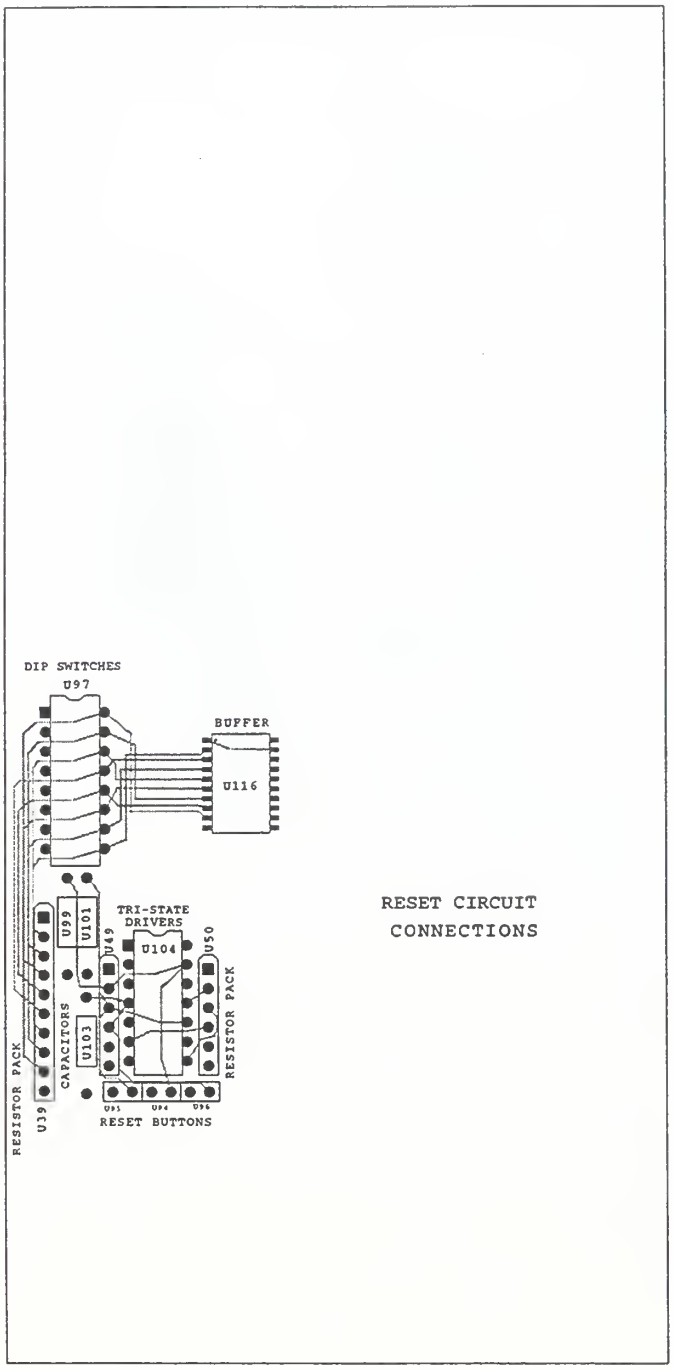



Figure 67. Reset circuit components

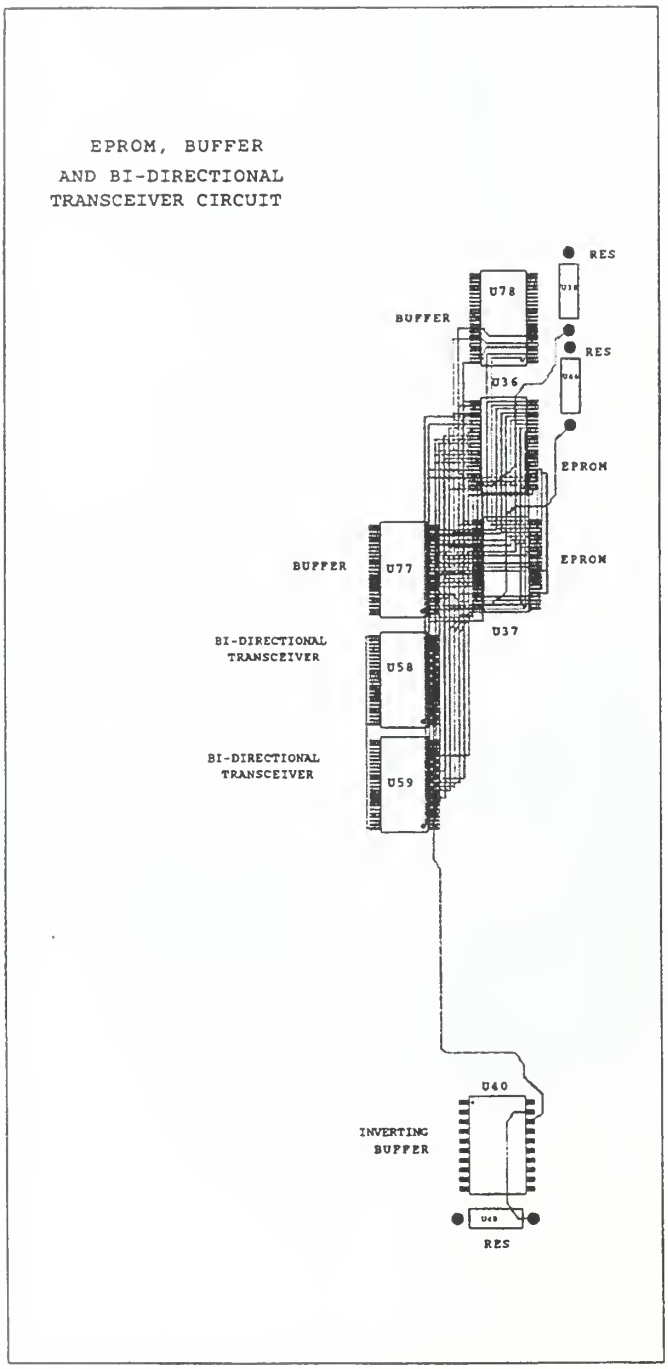



Figure 68. EPROM, buffer, and transceiver components

## APPENDIX D. TIMS PARTS LIST

| Item | Quantity | Reference | Part |
|---|---|---|---|
| 1 | 3 | U38 | 1K Resistor |
| | | U46 | 1K Resistor |
| | | U48 | 1K Resistor |
| 2 | 1 | U119 | 4.7K Resistor |
| 3 | 4 | U29 | 55K Resistor |
| | | U31 | 55K Resistor |
| | | U32 | 55K Resistor |
| | | U33 | 55K Resistor |
| 4 | 1 | U15 | 150K Resistor |
| 5 | 1 | U5 | 200K Resistor |
| 6 | 1 | U13 | 470K Resistor |
| 7 | 1 | U3 | 3.9M Resistor |
| 8 | 1 | U50 | 1K 5x1 Resistor Pack |
| 9 | 6 | U23 | 1K 9x1 Resistor Pack |
| | | U30 | 1K 9x1 Resistor Pack |
| | | U39 | 1K 9x1 Resistor Pack |
| | | U41 | 1K 9x1 Resistor Pack |
| | | U42 | 1K 9x1 Resistor Pack |
| | | U43 | 1K 9x1 Resistor Pack |
| 10 | 1 | U28 | 4.7K 5x1 Resistor Pack |
| 11 | 1 | U25 | 4.7K 9x1 Resistor Pack |
| 12 | 1 | U49 | 100K 5x1 Resistor Pack |
| 13 | 2 | U22 | 1.8µH Inductor |
| | | U24 | 1.8µH Inductor |

| Item | Quantity | Reference | Part |
|---|---|---|---|
| 14 | 2 | U1 | 20pF Capacitor |
| | | U2 | 20pF Capacitor |
| 15 | 2 | U10 | 0.1µF Capacitor |
| | | U19 | 0.1µF Capacitor |
| 16 | 2 | U20 | 0.22µF Capacitor |
| | | U21 | 0.22µF Capacitor |
| 17 | 3 | U99 | 1µF Capacitor |
| | | U101 | 1µF Capacitor |
| | | U103 | 1µF Capacitor |
| 18 | 1 | U8 | 10µF Capacitor |
| 19 | 1 | U9 | 100µF Capacitor |
| 20 | 1 | U11 | 1SMC5.0AT3 5.0V Diode |
| 21 | 2 | U14 | Green LED |
| | | U16 | Green LED |
| 22 | 2 | U97 | DIP Switch-8 |
| | | U80 | DIP Switch-8 |
| 23 | 3 | U94 | Hard Reset Switch |
| | | U95 | Soft Reset Switch |
| | | U96 | Abort Switch |
| 24 | 2 | U6 | Power Supply Jumper-3 |
| | | U18 | Battery Supply Jumper-3 |
| 25 | 2 | U51 | 74HC125 Tri-State Driver |
| | | U104 | 74HC125 Tri-State Driver |
| 26 | 1 | U40 | IDT 74FCT240 Inverting Buffer |
| 27 | 1 | U116 | IDT 74HC244 8-bit Buffer |
| 28 | 2 | U77 | IDT 74FCT16244 16-bit Buffer |
| | | U78 | IDT 74FCT16244 16-bit Buffer |

| Item | Quantity | Reference | Part |
|---|---|---|---|
| 29 | 2 | U58 | IDT 74FCT164245 Transceiver |
| | | U59 | IDT 74FCT164245 Transceiver |
| 30 | 1 | U120 | Sipex SP304 RS232/422 Driver/Receiver |
| 31 | 1 | U12 | LTI LT1086CM-3.3 DC-DC Converter |
| 32 | 1 | U17 | Maxim MAX704T Power Supervisor |
| 33 | 2 | U26 | Sharp LH28F016LL 16Mb Flash |
| | | U27 | Sharp LH28F016LL 16Mb Flash |
| 34 | 2 | U34 | EDI EDI8L32512C 16Mb SRAM |
| | | U35 | EDI EDI8L32512C 16Mb SRAM |
| 35 | 2 | U36 | Atmel AT27C800 8Mb UV EPROM |
| | | U37 | Atmel AT27C800 8Mb UV EPROM |
| 36 | 1 | U7 | Motorola MPC860 |
| 37 | 1 | U89 | DDC BU-61588 ACE BC/RT/MT |
| 38 | 1 | U45 | Beta Tech. DLP-7000 1553 Transformer |
| 39 | 1 | U4 | M-tron SX1555 32.768kHz Crystal |
| 40 | 2 | U117 | DB9 9-pin serial connector (Female) |
| | | U118 | DB9 9-pin serial connector (Female) |
| 41 | 1 | U44 | DB10 10-pin 1553 connector (Female) |
| 42 | 1 | U47 | DB8 8-pin debug connector (Female) |

## LIST OF REFERENCES


1. DMT Ettlinger, "Gannet picture," Shell Aviation News, United Kingdom.

2. McGrath, B., et al., "The Role of Intelligent Software in Spatial Awareness Displays," *Proceedings for the Third Annual Symposium and Exhibition on Situational Awareness in the Tactical Air Environment*, Piney Point, MD, June 2, 1998.

3. Rupert, A. and McGrath, B., Meeting with CDR Angus Rupert and Brad McGrath at the Naval Aerospace Medical Research Laboratory, June 20-21, 1997.

4. Gillingham, K. and Wolfe, J., "Spatial Orientation in Flight," USAF document USAFSAM-TR-85-31, USAF School of Aerospace Medicine, Brooks AFB, TX, December 1986.

5. McTrusty, T., "Swimmer Inshore Navigation System (SINS) Tactile Situation Awareness System (TSAS) Test Report," Coastal Systems Station, Panama City, FL, August 2, 1997.

6. Rochlis, J., "A Vibrotactile Display for Aiding Extravehicular Activity (EVA) Navigation in Space," Master's Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Boston, MA, 1998.

7. Clements, A., *Microprocessor Systems Design: 68000 Hardware, Software, and Interfacing*, 2nd edition, PWS Publishing Company, Boston, MA, 1992.

8. Weste, N., *Principles of VLSI Design, A Systems Perspective*, 2nd edition, Addison-Wesley Publishing Company, Reading, MA, 1993.

9. Wakerly, J., *Digital Design Principles and Practices*, 2nd edition, Prentice-Hall Inc., Englewood Cliffs, NJ, 1994.

10. Link, J., "Specification and Design of a Communication Protocol and Bus Interface Chip," Master's degree thesis work in progress, Naval Postgraduate School, Department of Computer and Electrical Engineering, graduating March, 1999.

11. Motorola Corporation, "MPC860 Application Development System Schematics," Revision B, Motorola Applications Engineering, February 13, 1997.

12. Motorola Corporation, "MPC860 Reference Design Board (SAMBA) Release Guide," Revision A, Motorola Applications Engineering, May 12, 1998.

13. Motorola Corporation, *MPC860 PowerQUICC User's Manual*, Version 1.0, Motorola Corporation, 1998.

14. Weicker R., *Dhrystone: A Synthetic Systems Programming Benchmark*, Communications of the ACM 27, October, 1984.

15. Aburto, A., *PDS: The Performance Database Server*, Naval Ocean Systems Center, San Diego, CA, 1998.

16. ILC Data Device Corporation, *MIL-STD-1553 Designer's Guide*, 4th ed., ILC Data Device Corporation, Bohemia, NY, 1993.

17. ILC Data Device Corp., "ACE Series BC/RT/MT Advanced Communication Engine Integrated 1553 Terminal User's Guide," Bohemia, NY, 1992.

18. Atmel Incorporated, *8-Megabit UV Erasable EPROM Data Sheet*, Atmel Corporation, 1997.

19. Sharp Electronics Corporation, *LH28F016LL 16M Flash Memory Data Sheet*, Sharp Electronics Corporation Microelectronics Group, Camas, WA, 1997.

20. Electronic Designs Incorporated, *512Kx32 CMOS High Speed Static RAM Data Sheet*, Electronic Designs Incorporated, Westborough, MA, 1997.

21. Motorola Corporation, *Some Characteristics and Design Notes for Crystal Feedback Oscillators*, Version 1.02, Motorola Applications Engineering, May, 1998.

22. Motorola Corporation, *MPC860 Board Design Checklist*, Version 1.02, Motorola Corporation, February, 1998.

## INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center ..... 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library ..... 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Chairman, Code EC ..... 1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

4. Professor Douglas J. Fouts, Code EC/Fs ..... 4
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

5. Randy L. Wight, Code EC/WT ..... 2
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

6. Angus H. Rupert, M.D., Ph.D. ..... 2
   CDR, Navy Medical Corps
   Naval Aerospace Medical Research Laboratory
   51 Hovey Road
   NAS Pensacola, FL 32508-1046

7. LT Brian Luke ..... 2
   5955 Osprey Place
   Pensacola, FL 32504